目　录

第1章 绪论

1.1 图像和图像处理

1.1.1 景象和图像

通常人眼所看到的客观存在的世界称之为景象。当我们从某一点观察某一景象时，物体所发出的光线（可以是发光物体本身辐射的光，也可以是物体受光源照射后反射或透射的光）进入人眼，在人眼的视网膜上成像。这个“像”反映了客观景物的亮度和颜色随空间位置和方向的变化，因此它是空间坐标的函数。视网膜成像是一种自然的生理现象，但只是到人类文明发展到一定时候人们才意识到它的存在，并设法用各种方法把它记录下来。这种记录下来的各种形式的“像”就是通常所说的图像，它包括各类图片、照片、绘画、文稿、X光胶片等等。这些图像都是人们对客观景象、事物以及人们的思维、想象的一种描述和记录，是人类用来表达和传递信息的一种重要手段。在人对外界信息的感知中，大约有70%是通过人的视觉系统，也就是以图像的形式获得的。而且随着科学技术的发展，人类不仅能获得并记录那些人眼可见的图像信息，即可见光范围内的图像，而且可以获得许多在通常情况下人眼无法看到的图像。这就是利用非可见光和其他手段成的“像”，并利用适当的换能装置将其变成人眼可见的图像，例如X射线成像、红外成像、超声成像、微波成像等等。这就使人的视觉能力大大地增强和延伸。总之，由于图像与人类活动关系非常密切，从日常家庭生活到生产、医疗、艺术、文教、军事、航天都离不开图像，因而对数字图像处理提出了越来越高的要求。正是这种迫切的实际需要，给图像处理技术的发展提供了巨大的动力，促进了图像处理技术的发展。

一幅平面图像所包含的信息首先表现为光的强度，它是随空间坐标(x, y)、光线的波长λ和时间t而变化的，因此图像函数可写成

$$I = f(x, y, \lambda, t) \tag{1.1-1}$$

按照不同的情况，图像可以分为各种类型。

若只考虑光的能量而不考虑它的波长时，在视觉效果上只有黑白深浅之分，而无色彩变化，这时称为黑白图像，图像函数可表示为

$$I = f(x, y, t) = \int_0^{\infty} f(x, y, \lambda, t) V_s(\lambda) \mathrm{d}\lambda \tag{1.1-2}$$

式中：$V_s(\lambda)$为相对视敏函数(见第 2 章)。

当考虑不同波长光的彩色效应时，则为彩色图像。根据三基色原理(见第 2 章)，任何一种彩色可以用红、绿、蓝三种基色来表示。所以彩色图像可表示成

$$I = \{f_r(x, y, t), f_g(x, y, t), f_b(x, y, t)\} \tag{1.1-3}$$

式中：

$$f_r(x, y, t) = \int_0^{\infty} f(x, y, \lambda, t) R_s(\lambda) \mathrm{d}\lambda$$

$$f_g(x, y, t) = \int_0^{\infty} f(x, y, \lambda, t) G_s(\lambda) \mathrm{d}\lambda$$

$$f_b(x, y, t) = \int_0^{\infty} f(x, y, \lambda, t) B_s(\lambda) \mathrm{d}\lambda$$

式中：$R_s(\lambda)$，$G_s(\lambda)$，$B_s(\lambda)$分别是红、绿、蓝三基色的视敏函数。与此相应的还有由不同光波段的光形成的多谱图像。

当图像内容随时间变化时，称之为时变图像或运动图像，它的图像函数如式(1.1-1)～(1.1-3)所示。运动目标的图像，电影，电视的画面等都是运动图像。

当图像内容不随时间变化时，称之为静止图像。对黑白图像而言，静止图像函数为

$$I = f(x, y) \tag{1.1-4}$$

由于人眼的视野是有限的，因此图像在空间上是有界的，而且通常定义为矩形，即

$$0 \leqslant x \leqslant L_x, \ 0 \leqslant y \leqslant L_y \tag{1.1-5}$$

图像函数在某一点的值通常称之为强度或灰度，与图像在这一点的亮度(Brightness)相对应，并用正实数表示，而且这个数值的大小是有限的。通常约定，图像灰度值大、表示亮，反之则表示暗，即

$$0 \leqslant f(x, y) \leqslant B_m \tag{1.1-6}$$

B_m 为最大亮度。所以图像函数 $f(x, y)$是一个二元、有界、非负的连续函数。

上面讨论的人眼能够看到的图像称为模拟图像，它的函数是连续的、可解析的，因而是可积的，有可逆的傅里叶变换等。但是，计算机无法接受模拟形式的图像，只有将模拟图像转换成数字图像才能用计算机进行处理。这种转换就是图像的数字

化,将在第2章中详细讨论。简略地说,一幅模拟图像经过采样和量化使其在空间上和数值上都离散化,形成一个数字点阵,通常采用等间隔采样和均匀量化。

这样,一幅静止的黑白图像经过数字化以后可以用一个大小为 $M\times N$ 的矩阵 $[f(m,n)]$ 来表示。这里 $m=0,\cdots,M-1$, $n=0,\cdots,N-1$, M, N 为图像的总行数和总列数,$f(m,n)$ 表示第 m 行 n 列像素(Pixel或简称Pel)的灰度值。它是一个整数,例如若用1 Byte存储一个像素,其值在0~255之间。有时为了表示方便,就用 $f(m,n)$ 来表示一幅经数字化后的黑白静止图像。相应的,彩色图像经数字化以后可以用三个矩阵来表示,它们是 $[f_r(m,n)]$, $[f_g(m,n)]$ 和 $[f_b(m,n)]$,分别表示彩色图像的红、绿、蓝三个分量。

用于遥感的多谱图像经数字化后可用 K 个矩阵 $[f_k(m,n)]$, $k=1,\cdots,K$ 来表示,这里 K 表示光波段的总数。例如:NASA的LANDSAT卫星的多光谱图像有7个分量($K=7$), $k=1$ 对应于蓝光波段。$k=2$ 绿光, $k=3$ 红光, $k=4,\cdots,7$ 对应于不同波长的红外光波段。利用多谱图像进行地面物体的识别是相当有效的技术手段。

在计算机层析仪(CT)中,则用 $[f_k(m,n)]$ 来表示第 k 个截面的截面图像,它反映了不同位置处被测物体对X光吸收程度之强弱。

在数字电视或动态图像分析中,需要进行序列图像处理。序列图像可用 $[f_k(m,n)]$ 来表示,这里 k 为帧序号,表示时间轴上的采样。彩色活动图像则要用三个分量的序列来表示,即

$$[f_{rk}(m,n)],\ [f_{gk}(m,n)],\ [f_{bk}(m,n)],\ k=1,2,\cdots,$$

在一般的视频(Video)应用中,常采用别的彩色坐标系,如可以用 Y, C_b, C_r 即亮度与色差来表示。则此时 $[f_{Yk}(m,n)]$, $[f_{C_bk}(m,n)]$, $[f_{C_rk}(m,n)]$ 三个图像序列用来表示一段时间里的彩色活动图像。

1.1.2 数字图像处理的内容和范畴

图像处理方法可分为两大类,即光学图像处理和数字图像处理。当然也有将光学和数字相结合的混合图像处理。

光学图像处理就是用光学的方法对模拟图像作某种处理。如:用光学傅里叶透镜对图像进行二维傅里叶变换等处理。光学图像处理是一种典型的模拟图像处理,它不在本书讨论范围以内。

数字图像处理就是用计算机对各种图像信息进行处理,以期得到某种预期的效果或从图像中提取有用的信息。这里的"计算机"是广义的,它不仅指通用计算机,也指由DSP,各种专用处理芯片(ASIC)等构成的通用或专用处理系统。而这里的"处理"则有广义和狭义的不同理解,它涉及的内容包括了图像处理,模式识别

和计算机视觉中的许多概念和方法。根据抽象程度的不同可把这些处理分为低、中、高三个层次。低层处理涉及对图像进行加工以改善图像的视觉效果，或对图像数据进行压缩以利于图像的存储和传输。典型的处理方法如：图像对比度增强、边缘锐化、去噪声等。这些都是典型的图像增强处理，处理后的图像可以使人看起来更清楚。对散焦图像或因运动造成的模糊图像的处理则是要去除因散焦或因相机与物体间的相对运动造成的模糊，通常用图像复原技术进行处理。而为了节省传输带宽和存储空间，需要对图像进行压缩编码。前述各种处理都是从原始图像到处理后图像，即从图像到图像的处理过程。为了得到好的视觉效果，需要对人眼视觉系统(Human Visual System, HVS)的特点有深入的了解，而这正是难点所在。

中层处理主要是指用某种特殊手段提取、描述和分析图像中所包含的某些特征或特殊的信息，主要目的是便于计算机对图像做进一步的分析和理解，经常作为模式识别、计算机视觉等的预处理。这些特征包括很多方面，例如，图像的频域特征、灰度特征、边界特征、颜色特征、纹理特征、形状特征、拓扑特征以及关系结构等。这种处理过程往往首先要进行图像分割，把感兴趣的对象(Object，有些场合也称之为目标)从图像中分割出来，然后对它的特征进行测量，用特征数据来表示它，或用一些符号来表示对象或多个对象间的关系。这就是图像分割和描述。这里，输入的是图像；输出的则是数据，而不是图像。有些看似简单的图像要进行理想的分割和描述并不容易，这里涉及人对认知机理的了解。

高层处理涉及在图像分析中被识别物体的总体理解。通过对图像内容的理解及对场景的解释，进而指导和规划行动。

这三个层次的处理之间的界限有时并不清晰，它们之间有相互联系、相互依存的关系。下层处理结果不好会影响上层的正确识别和理解。反之，若在下层处理时能有效利用上层识别理解的中间结果，则可大大改进下层的处理结果。本书的内容限制在“狭义”的数字图像处理范畴以内，即包括图像增强、复原、压缩、分割和描述等，而不包括识别和理解，后者是模式识别和计算机视觉的经典内容。

1.1.3 数字图像处理发展历史和应用领域

数字图像处理的最早应用要追溯到20世纪20年代，当时图像处理技术首次应用于图像的远距离传送，用来改善英国伦敦和美国纽约之间经海底电缆发送的图片质量。当时世界上第一台计算机尚未诞生，采用的是一种用编码图像纸带去调制光束而使底片感光的系统。随着计算机和集成电路的诞生和飞速发展，数字图像处理才真正引起人们的巨大兴趣，并得到迅速发展。用计算机进行图像处理、改善图像质量的有效应用始于1964年，美国喷气推进实验室(J. P. L.)用计算机对“徘徊者7号”、“水手4号”、阿波罗载人登月飞船等传回的图片都用了数字图像处

理技术进行处理，当时的主要处理是图像的校正、增强、复原等。1972年，美国陆地资源卫星(Landsat)将多谱图像用于资源探测。遥感图像处理的应用面非常广，如：地形地质探测，森林、水利、海洋、农业等资源调查，自然灾害的预测和预报，环境污染的监测，气象卫星云图的处理等。

1972年计算机层析仪(CT，或称CAT)问世。它将图像重建技术用于医学，取得了巨大成功。更新的技术是核磁共振成像(MRI)。其他著名的医学应用有数字减影技术(DSA)、染色体分类、血球分类技术等。

数字图像处理在工业生产中的应用范围十分广泛。在产品的质量检测方面，如无损探伤图像处理，可以检查零部件内部的缺损、焊缝质量，进行金属材料的成分和结构的分析。在工业自动控制中可以进行纺织品质量的检查，监视零部件的装配等。在港口的检测调度，交通管理，车型、车牌识别等应用领域均可采用图像处理技术。

在军事、公安等应用领域，图像处理可用于目标的侦察，目标识别跟踪制导，如基于图像匹配的巡航导弹等。在公安、安全部门，有人脸识别，指纹识别和指纹库，图像监控等。

目前，图像压缩编码技术已经得到了广泛的推广应用，对人们的日常生活产生了重要的影响。VCD，DVD和各种档次的数字电视，如标清电视、高清电视、数字电影、网络电视及即将到来的手机电视等都是图像压缩技术的应用实例。

冈萨雷斯教授在他的书中写道："今天，几乎不存在与数字图像处理无关的领域。"

1.2 数字图像处理系统简介

数字图像处理主要依靠计算机对图像数据进行加工。但是在处理以前首先必须把图像信息转换成数据送入计算机，因此必须有图像输入和数字化设备。图像处理完毕后必须把它显示出来或记录成硬拷贝，这就需要有显示及记录设备。所以，一般说来，数字图像处理系统由三大部分组成，即图像输入及数字化设备，图像信息处理设备和图像显示及记录设备，如图1.1所示。

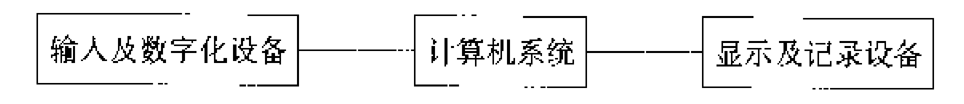


图1.1 数字图像处理系统的三大组成部分

图像信息的处理由计算机担任。由于数字图像信息量十分巨大，要求计算机系统运算速度快，存储容量大(包括内存及外存)，且有较强的软件功能。根据图像处理系统不同的用途，可以采用不同的计算机系统，从微型计算机到大型计算机；可以是单个计算机，也可以用阵列机、多处理机或计算机网络。专用的图像处理系

统的信息处理常常用一片或几片专用芯片来完成。

图像输入设备可以采用电视摄像机、鼓式扫描器或固体器件等等。它们在输入信息的速度、空间分辨率(扫描精度)等方面各有特点,可以根据需要来选用。一个通用的数字图像处理系统也可同时配有几种图像输入设备,以适应不同的需要。图像输入设备一般只起光电转换的作用,即将图像的光强信息转变为模拟电信号。然后,再送到数字化设备,即模数转换器转变为数字图像。模数转换器完成采样和量化两个过程,一般由专用芯片来完成。有些图像输入设备输出的已是数字信号,则可经数字接口直接接入计算机。

经过处理以后的图像可以在图像显示器上显示,同时增设专门的交互式控制设备,如数字化仪(Tablet)、鼠标、跟踪球、游戏杆、光笔等。通过这些设备可以把人的作用反馈给计算机,干预和引导计算机的运行,使系统的处理能力大大提高。


图 1.2 是一个通用数字图像处理系统的方框图。

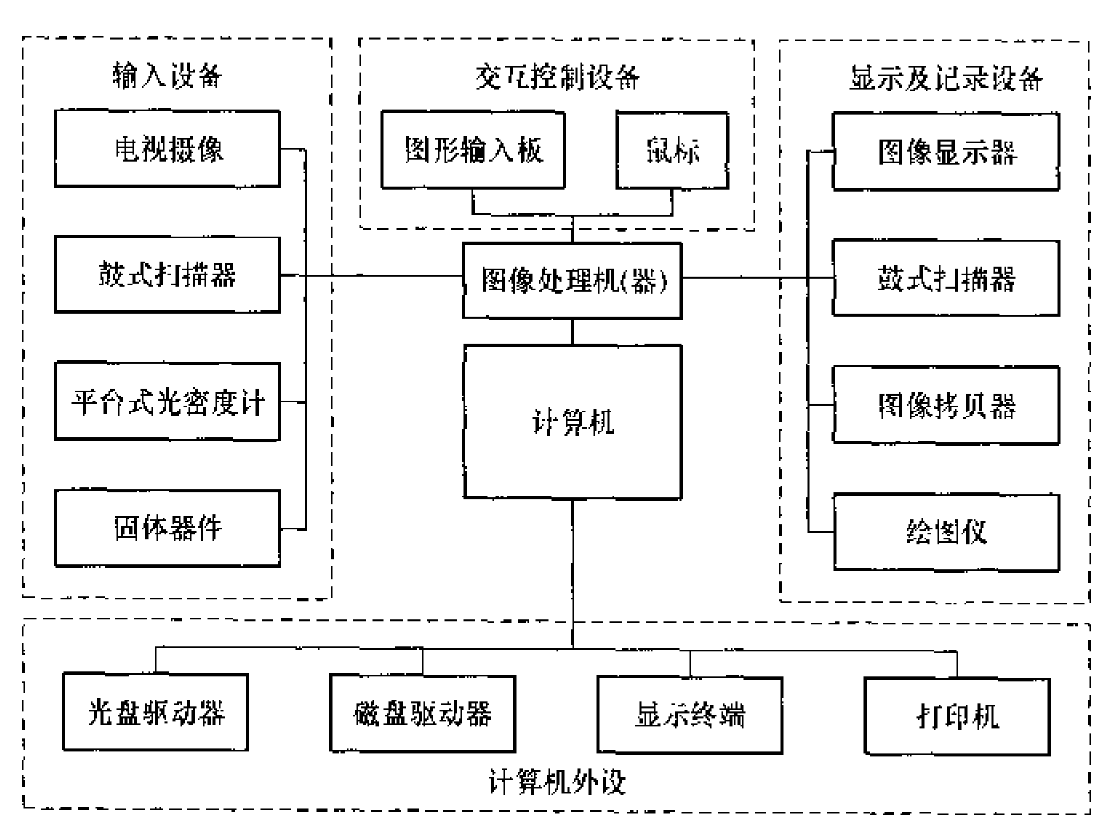


图 1.2 通用数字图像处理系统的组成框图

1.3 本书的内容安排

数字图像处理的内容十分丰富,应用范围十分广泛。本书的内容限制在通常意义的,即狭义的数字图像处理范围内。对诸如图像内容的识别、理解等内容感兴

趣的读者可参阅模式识别、计算机视觉方面的有关书籍。

全书分三个部分共8章和一个附录。第一部分是图像处理基础(1,2,3章)。第二部分介绍与视觉相关的几种常用图像处理技术,包括图像增强、复原、重建、压缩编码(4,5,6章)。第三部分介绍与机器识别理解有关的图像处理技术,包括图像的分割与描述(7,8章)。附录则对数学形态学的原理作简介。各章内容如下:

第1章 绪论

它对图像处理技术作概括性的介绍。对各种用途图像信息的来源、表示方法、数字图像处理的内容和范畴、发展历史和应用领域作概括的介绍。并对通用数字图像处理系统的构成作简要说明。

第2章 数字图像处理基础

它对图像的数字化、光度学、色度学基础、视觉模型和视觉特性、图像质量的主客观评价等内容作了介绍。

第3章 图像的二维正交变换

它对二维正交变换的数字表达、图像的离散傅里叶变换、离散余弦变换、离散卡-洛变换、沃尔什-哈达玛变换、哈尔变换等内容作了较详细的讨论。并对近年来在图像处理中用得越来越多的小波变换作了介绍,着重从概念上对小波变换概念和方法进行讨论。

第4章 图像增强

介绍各种常用的图像增强技术。包括图像的灰度变换、去噪声、各种空间域或变换域滤波技术、同态滤波、中值滤波、小波去噪及彩色图像增强。图像增强是提高图像视觉质量的有效方法,是简单有效的图像处理技术。怎样用简单的处理提高图像的视觉质量,是本章讨论的核心问题。

第5章 图像复原与重建

图像复原是通过处理将降质图像进行复原,尽量恢复原图的过程。本章主要通过代数方法对常用的无约束和有约束复原技术进行讨论。本章还对由投影重建截面的图像重建技术的原理进行了讨论。

第6章 图像数据压缩编码

图像压缩编码技术是目前应用最广泛的图像处理技术。本章介绍图像编码的基本原理和常用技术,包括预测编码、变换编码、各种熵编码技术。本章对静止图像和活动图像编码的原理和框架作了介绍。对近年来的各种图像编码国际标准作了简介,对这些标准的具体规定有兴趣的读者,可参阅有关的标准文本和相关文献。怎样有效地将信息论和视觉特性用于图像编码,是本章讨论的核心问题。

第7章 图像分割

本章介绍各种常用的图像分割技术。图像分割是机器对图像内容进行识别理

解的第一步,也是十分关键的一步。目前没有一种适用于各种应用的通用图像分割算法,需要使用者根据应用目的选择合适的方法和参数。本章介绍了较多的分割方法,对其原理和步骤一一介绍,这样做是为了使读者在今后实用中根据需要进行选择和改进。图像分割的方法可分为二大类:基于边缘的分割和基于区域的分割,在基于边缘的分割方法中,除了介绍拉普拉斯、LOG,canny 算子等边缘检测算子外,还对 snake 轮廓跟踪技术和霍夫变换作了介绍。在基于区域的分割方法中对阈值化、区域生长和区域分裂合并法作了介绍,并介绍了分水岭法。本章还对分割算法的评价作了简介。

第 8 章　图像描述

本章介绍对图像分割所提取的各种感兴趣的对象进行描述的方法。通过对这些对象的形状、属性和相互关系进行合适的描述,可以有效地利用它们。图像描述技术可分为针对边界(线)的描述和针对区域的描述两大类。在线的描述方面,介绍了曲线拟合、链码、傅里叶描述子和小波描述子等。在区域描述方面,介绍了中轴变换、四叉树、拓扑描述子等方法。本章最后介绍了各种特征测量的方法。

附录　数学形态学简介

数学形态学是图像分析的重要数学工具,它特别适合于并行处理机的处理。附录主要对二值图像的数学形态学基本运算(腐蚀、膨胀、结构开、结构闭等)的定义和性质作了介绍,并将其推广到灰度图像的数学形态学处理。

参 考 文 献

[1] 冈萨雷斯. 数字图像处理(第二版)[M]. 阮秋琦,等译. 北京:电子工业出版社,2003.

[2] Kenneth. R. Castleman. 数字图像处理[M]. 北京:电子工业出版社,1998.

[3] 章毓晋. 图像处理和分析(图像工程上册)[M]. 北京:清华大学出版社,1999.

[4] 阮秋琦. 数字图像处理学[M]. 北京:电子工业出版社,2001.

[5] 赵荣椿. 数字图像处理导论[M]. 西安:西北工业大学出版社,1996.

[6] Yao Wang,等. 视频处理与通信[M]. 侯正信,等译. 北京:电子工业出版社,2003.

[7] Ed. Al Bovik. Handbook of Image and Video Processing [M]. Academic Press, 2000.

[8] Murat Tekalp A. Digital Video Processing [M]. Prentice Hall, 1995.

[9] Ed. H. R. Wu, Rao, K. R., Digital Video Image Quality and Perceptual Coding [M]. CRC Tayler & Francis, 2006.

[10] Yun Q. Shi, Huifang Sun. Image and Video Compression for Multimedia Engineering, Fundamentals, Algorithms and Standards [M]. CRC Press, 2000.

第2章 数字图像处理基础

图像以数字形式进入计算机，是进行数字图像处理的第一步。一幅黑白图像(单色图像)可以看成一个二维连续函数 $f(x, y)$，其亮度表现为位置(x, y)的连续函数。计算机中的数字图像是以矩阵形式$[f(m, n)]$表示的。不同的处理就是用不同的算法对这一图像矩阵进行运算。从二维连续函数到矩阵，涉及在不同空间位置上取出函数(亮度)值作为样本，并用一组整数值来表示这些样本的两个过程。前者称为取样，后者称为量化，二者统称为数字化。经数字化后得到的图像$[f(m, n)]$能否保持 $f(x, y)$的原有信息，即在空间上对 $f(x, y)$的取样密度多大才是合适的，在幅度上以多少等级表示样本的亮度(以下称为像素的灰度)才足够，然后用怎样一种方法才能从$[f(m, n)]$恢复出 $f(x, y)$，将是本章所要讨论的问题。

本章将介绍光的一些基本特性及其度量，介绍色度学方面的有关内容。这些知识对于更好地掌握图像处理这门学科是十分必要的。

数字图像处理的一个任务是恢复受损图像的本来面貌，或用专门的方法突出图像中某些特定的目标，由人眼判读。这种判读与人眼主观视觉感受密切有关。为了使图像处理的算法更有效、更有针对性，在某些场合中利用或避开人眼主观视觉上的长处或局限，有必要了解一下人眼主观视觉上的机理及特点。限于本书的性质及篇幅，本章仅对此作一些必要的介绍。

随着数字图像处理技术近年来在许多领域的广泛应用，各种图像感知质量尺度伴随着人眼视觉模型不断被提出来。本章将对人眼视觉模型和图像质量的主客观评价方法进行介绍。

2.1 取样与重建

取样是图像 $f(x, y)$进入计算机的第一个处理过程。事实上在数字图像处理学诞生之前，对某些图像已经采用取样技术进行处理了。为了传输传真图像和电视图像，需要在扫描鼓上逐行读取图像信息，或在摄像管上由扫描光栅逐行获取图像信息。经过光电转换的过程，把图像信号以点的形式送到后续设备中去。上述过程中，仅仅在 x 方向上完成了对图像的取样。在数字图像中还要沿 y 方向进行取样。对图

像 $f(x, y)$沿 x 方向取样 M 点，沿 y 方向取样 N 点，便得到了矩阵$[f_s(m, n)]_{M\times N}$。

我们要研究的主要问题是 $f_s(x, y)$的取样密度应当是多大，才不至于丢失原图的信息。不丢失信息而能完整地恢复原图是对取样的基本要求。本节所介绍的惠特克-卡切尼柯夫-香农(Whittaker-Kotelnikov-Shannon)取样定理将回答取样所应遵循的准则。

取样是决定一幅图像空间分辨率的主要参数，图 2.1 中显示了一幅空间分辨率为 256×256 的图像经不同比例取样后所达到的效果。

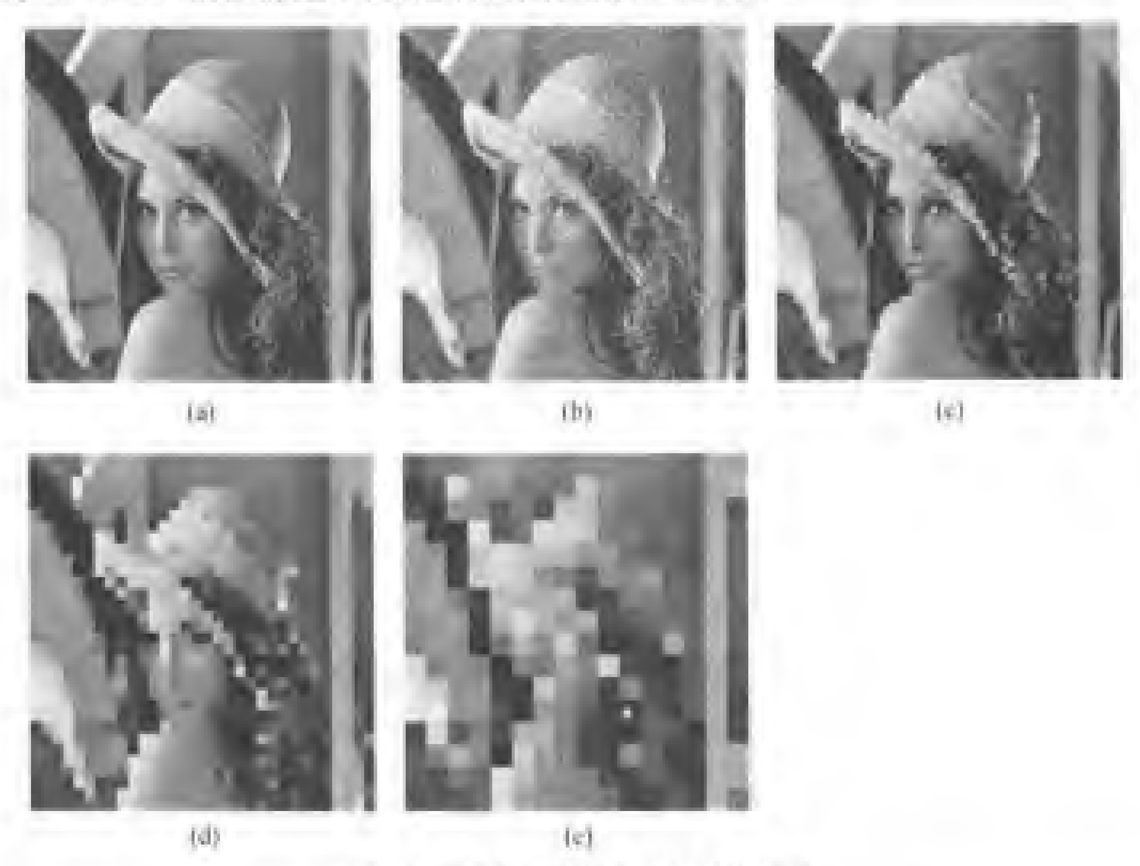


图 2.1 不同空间分辨率的图像效果

(a)为 256×256，8 比特图像；(b)～(e)为 128×128，64×64，32×32，16×16 图像放大到 256×256 像素的图像

2.1.1 取样

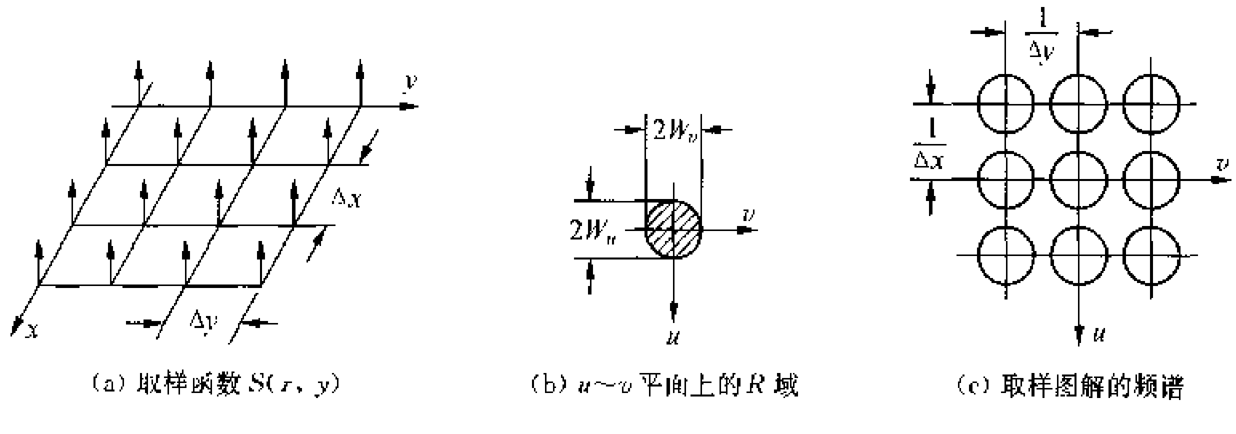
假定有一个二维连续函数 $f(x, y)$，$x, y\in(-\infty, +\infty)$，它是一个有限带宽的函数。设 $F(u, v)=\mathscr{F}\{f(x, y)\}$，则应当有：

当 $u\notin[-W_u, +W_u]$ 或 $v\notin[-W_v, +W_v]$ 时，

$$F(u, v)=0 \tag{2.1-1}$$

式中：W_u 和 W_v 分别表示在 u 和 v 方向上的带宽，这就是说在傅里叶变换域(uv 域)中，原图 $f(x, y)$的信息集中在长宽为 $2W_u$ 和 $2W_v$ 的矩形范围的 R 域中，见图 2.2(b)。

有一个二维取样函数 $s(x, y)=\sum_{m=-\infty}^{+\infty}\sum_{n=-\infty}^{+\infty}\delta(x-m\Delta x, y-n\Delta y)$，它是沿 x 方向间隔为 Δx，沿 y 方向间隔为 Δy 的二维冲激函数阵列，见图 2.2(a)。



(a) 取样函数 $S(x, y)$　　(b) $u \sim v$ 平面上的 R 域　　(c) 取样图解的频谱

图 2.2　二维取样图解

用 $s(x, y)$ 对 $f(x, y)$ 取样，得到取样图像 $f_s(x, y)$，它可以表示为强度是图像样本灰度值的二维冲激函数阵列，即：

$$f_s(x, y)=s(x, y)f(x, y)=\sum_{m=-\infty}^{+\infty}\sum_{n=-\infty}^{+\infty}f(x, y)\delta(x-m\Delta x, y-n\Delta y)$$

$$=\sum_{m=-\infty}^{+\infty}\sum_{n=-\infty}^{+\infty}f(m\Delta x, n\Delta y)\delta(x-m\Delta x, y-n\Delta y) \tag{2.1-2}$$

这一步相当于对 $f(x, y)$ 以矩形点阵均匀取样，每个取样位置在 $x=m\Delta x$，$y=n\Delta y$ 上，其中，$m, n=0, \pm 1, \pm 2, \cdots$。

现在来看取样后 $f(x, y)$ 的谱 $F(u, v)$ 所发生的变化。可以推导出 $s(x, y)$ 的傅里叶变换式为：

$$S(u, v)=\mathscr{F}\{s(x, y)\}$$

$$=\frac{1}{\Delta x}\frac{1}{\Delta y}\sum_{m=-\infty}^{+\infty}\sum_{n=-\infty}^{+\infty}\delta\left(u-m\frac{1}{\Delta x}, v-n\frac{1}{\Delta y}\right) \tag{2.1-3}$$

现 $f_s(x, y)=s(x, y)f(x, y)$，因此它的傅里叶变换式 $F_s(u, v)$ 应当是 $S(u, v)$ 和 $F(u, v)$ 之间的卷积：

$$F_s(u, v)=\mathscr{F}\{f_s(x, y)\}=\mathscr{F}\{s(x, y)f(x, y)\}$$

$$=S(u, v)*F(u, v)$$

$$=\iint_{-\infty}^{+\infty}\frac{1}{\Delta x}\frac{1}{\Delta y}\sum_{m=-\infty}^{+\infty}\sum_{n=-\infty}^{+\infty}\delta\left(\alpha-m\frac{1}{\Delta x}, \beta-n\frac{1}{\Delta y}\right)F(u-\alpha, v-\beta)\mathrm{d}\alpha\mathrm{d}\beta$$

$$= \frac{1}{\Delta x}\frac{1}{\Delta y}\sum_{m=-\infty}^{+\infty}\sum_{n=-\infty}^{+\infty}\iint_{-\infty}^{+\infty}\delta\left(\alpha - m\frac{1}{\Delta x},\ \beta - n\frac{1}{\Delta y}\right)F(u-\alpha,\ v-\beta)\mathrm{d}\alpha\mathrm{d}\beta$$

$$= \frac{1}{\Delta x}\frac{1}{\Delta y}\sum_{m=-\infty}^{+\infty}\sum_{n=-\infty}^{+\infty}F\left(u - m\frac{1}{\Delta x},\ v - n\frac{1}{\Delta y}\right) \tag{2.1-4}$$

式(2.1-4)表明已取样图像 $f_s(x,\ y)$ 的谱是原来 $f(x,\ y)$的谱 $F(u,\ v)$沿 u 轴和 v 轴分别以$\frac{1}{\Delta x}$和$\frac{1}{\Delta y}$无限地周期性重复的结果，见图 2.2(c)。

用一个理想的二维低通滤波器，其频率响应为

$$H(u,\ v) = \begin{cases} 1, & u \in [-W_u,\ W_u] \text{ 且 } v \in [-W_v,\ W_v] \\ 0, & \text{其他} \end{cases} \tag{2.1-5}$$

把 $H(u,\ v)$与 $F_s(u,\ v)$相乘，即可取出原来的 $F(u,\ v)$，经反变换后可以完整地得到 $f(x,\ y)$。

从图 2.2(c)中可以看出，当 $f(x,\ y)$是有限带宽，即满足(2.1-1)式，且同时满足

$$\begin{cases} \dfrac{1}{\Delta x} \geqslant 2W_u \\ \dfrac{1}{\Delta y} \geqslant 2W_v \end{cases} \quad \text{即} \quad \begin{cases} \Delta x \leqslant \dfrac{1}{2W_u} \\ \Delta y \leqslant \dfrac{1}{2W_v} \end{cases} \tag{2.1-6}$$

则各个相邻的 R 域不会彼此混叠，可以用一个理想的低通滤波器取出一个完整的 R 域，以确保取样后不失真地再现原信号 $f(x,\ y)$，这就是二维取样定理。从式(2.1-6)可以看出，这里的取样是以矩形的点阵均匀取样，两个方向上的取样间隔分别与两个方向上的信号带宽有关。

$f_s(x,\ y)$ 仅仅在 $x = m\Delta x,\ y = n\Delta y$ 的位置上有取样值，其他地方均为 0，为书写得更明确，可把 $f_s(x,\ y)$改写为 $f_s(m,\ n)$。$f_s(m,\ n)$构成了离散阵列，只是它的函数值还是在一个实数范围内，即还未被量化。

如果 $f(x,\ y)$ 在 $x \in [0,\ X]$，$y \in [0,\ Y]$ 内有定义，并以 Δx 和 Δy 为间隔取样，则沿 x 方向和 y 方向的取样点数分别为

$$M = \frac{X}{\Delta x} + 1,\ N = \frac{Y}{\Delta y} + 1$$

$f_s(x,\ y)$构成一个 $M \times N$ 的实数矩阵，写成$[f_s(m,\ n)]_{M\times N}$。

在以上的讨论中，假定采用了在空间位置上均匀取样的方法，在二维取样中取样点阵是矩形的或者是正方形的。这些方案在实践中容易实现，也便于分析。

取样定理在图像数字化过程中确保信息不丢失方面起着理论上的保证作用，

它从一个方面保证了其后的数字图像处理能取得最好的效果。在任何实际的数字图像处理系统的设计及实现中,满足取样定理是实现主要技术指标极为关键的一步。由于图像的内容千变万化,获取图像 $f(x, y)$ 的方法也是多种多样的,而取样定理是否得到了满足往往关系到后续处理的成败和质量。

在数字信号处理和数字图像处理一些理论问题的推导方面有一个有趣的问题,这里简单提及一下。上述结论是假定 $f(x, y)$ 在空间域 (x, y) 内无限延续的条件下得出的,且进行了无穷个取样。参考文献[1]指出,实际情况下 $f(x, y)$ 是在有限区间上存在的函数。这时应人为地假定在整个 (x, y) 域内,$f(x, y)$ 是一个周期函数,而只需在一个周期内进行有限的取样。只要取样定理得到满足,就可根据这些有限的取样值,把这个假想的周期函数完整地恢复出来,然后取其中的一个周期就是原来的 $f(x, y)$。这是因为任何有限延续的函数都不是有限带宽的;一个有限带宽的函数在 (x, y) 域中必然从 $-\infty$ 连续到 $+\infty$。这个结论在某些问题的推导中不可忽视。

2.1.2 重建

图像的重建是取样过程的逆过程,它完成从已取样图像 $f_s(m, n)$ 到连续图像 $f(x, y)$ 的变换。通过空间滤波或者空间内插就可以完成重建过程。

上节指出,当 $f(x, y)$ 是有限带宽,并且取样间隔 Δx 和 Δy 满足式(2.1-6)时,已取样图像 $f_s(m, n)$ 的谱 $F_s(u, v)$ 是原图 $f(x, y)$ 的谱 $F(u, v)$ 沿 u、v 轴以间隔 $\frac{1}{\Delta x}$ 和 $\frac{1}{\Delta y}$ 周期性的重复。此时若有一个理想的二维低通滤波器,其频率特性为 $H(u, v)$,见式(2.1-5),即可取出 $F(u, v)$,写作:

$$F(u, v) = kH(u, v) \cdot F_s(u, v) \tag{2.1-7}$$

即

$$f(x, y) = kh(x, y) * f_s(m, n) \tag{2.1-8}$$

式中:k 为比例因子,这里 $k = \Delta x \Delta y$。上式表明,从计算机中得到 $f_s(m, n)$ 后,只要与 $kh(x, y)$ 作卷积,就可不失真地重现 $f(x, y)$。

从图2.2(b)可见,原图 $f(x, y)$ 的信息集中在长宽为 $2W_u$ 和 $2W_v$ 的矩形范围内的 R 域中。若假定 $\Delta x = \frac{1}{2W_u}$,$\Delta y = \frac{1}{2W_v}$,则

$$\begin{aligned} kh(x, y) &= \mathscr{F}^{-1}\{kH(u, v)\} = \frac{\sin 2\pi W_u x}{2\pi W_u x} \frac{\sin 2\pi W_v y}{2\pi W_v y} \\ &= \sin c(2\pi W_u x) \sin c(2\pi W_v y) \end{aligned} \tag{2.1-9}$$

由式(2.1-2)，已取样图像可以看成强度为 $f(m\Delta x, n\Delta y)$ 的二维冲激函数阵列，即：

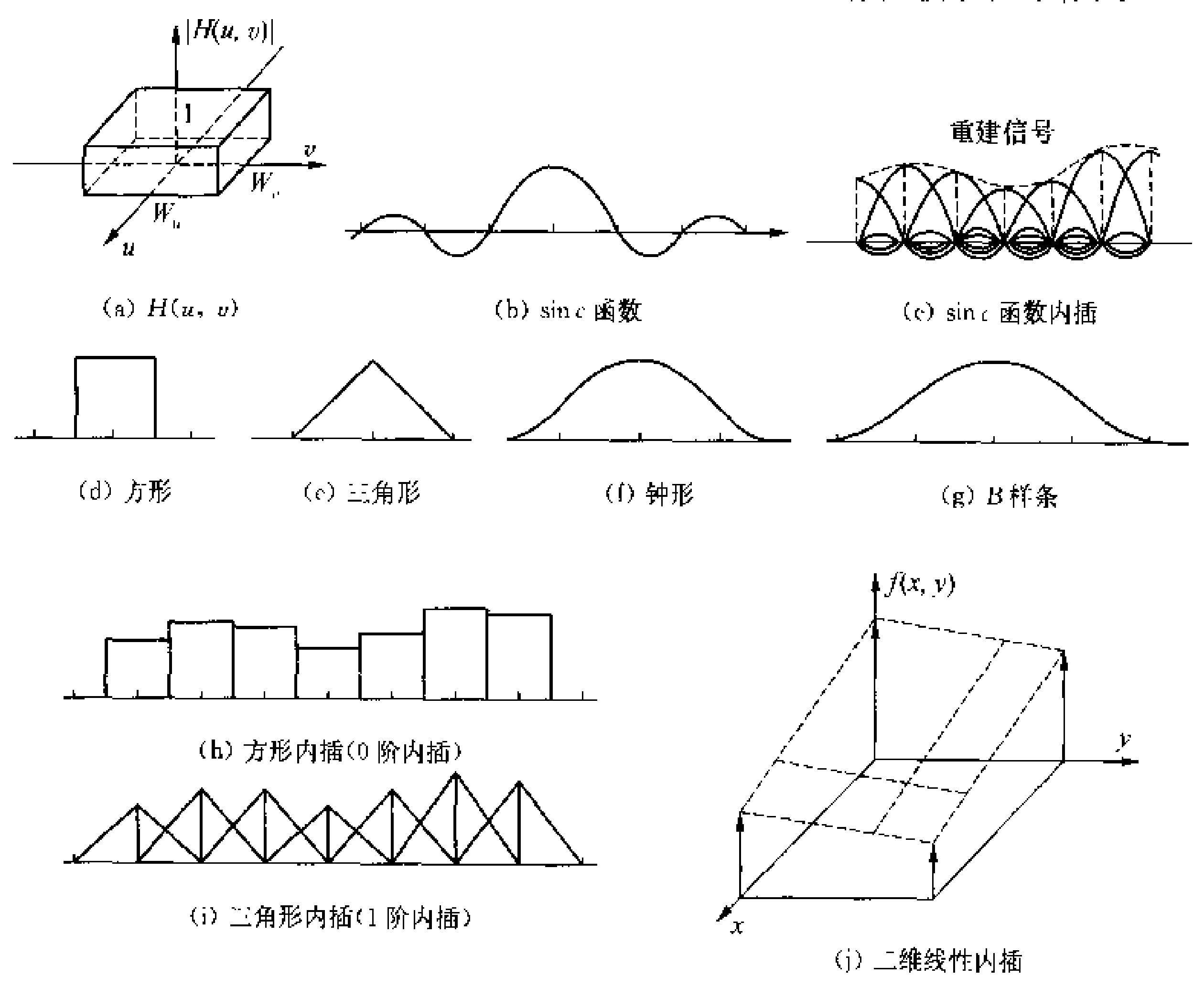
$$f_s(m, n) = \sum_m \sum_n f(m\Delta x, n\Delta y)\delta(x - m\Delta x, y - n\Delta y) \quad (2.1-10)$$

把式(2.1-9)和(2.1-10)代入式(2.1-8)，有：

$$\begin{aligned} f(x, y) &= \sum_m \sum_n \iint_{-\infty}^{\infty} \sin c[2\pi W_u(x-\alpha)]\sin c[2\pi W_v(y-\beta)] \\ &\quad \cdot f(m\Delta x, n\Delta y)\delta(\alpha - m\Delta x, \beta - n\Delta y)\mathrm{d}\alpha\mathrm{d}\beta \\ &= \sum_m \sum_n \sin c[2\pi W_u(x - m\Delta x)]\sin c[2\pi W_v(y - n\Delta y)]f(m\Delta x, n\Delta y) \end{aligned} \quad (2.1-11)$$

上式表明：重建图像是位于 $x = m\Delta x$，$y = n\Delta y$ 上的许多个二维 $\sin c$ 函数加权求和的结果，而加权值就是取样图像的值(即像素值)。

图 2.3(b)是一维 $\sin c$ 函数，(c)是一维情况下 $\sin c$ 函数加权求和的结果。



(a) $H(u, v)$　(b) $\sin c$ 函数　(c) $\sin c$ 函数内插

(d) 方形　(e) 三角形　(f) 钟形　(g) B样条

(h) 方形内插(0 阶内插)

(i) 三角形内插(1 阶内插)

(j) 二维线性内插

图 2.3　图像重建的图解

由 $\sin c$ 函数进行图像重建可以得到理想的结果，但是 $\sin c$ 函数在部分区间为负值，由它进行内插，技术上有一定的困难。事实上常采用其他类型的内插函数来替代 $\sin c$ 函数。

图 2.3 中示出部分内插函数的形状。其中(d)是方形函数，可得到 0 阶样本内插；(e)是三角形函数，可得到 1 阶样本内插(线性内插)，三角形函数可视作两个方形函数的卷积；(f)是钟形函数，它是三个方形函数卷积的结果；(g)是 B 样条函数，它是四个方形函数卷积的结果，其两端具有连续性和平滑性。图 2.3(h)和(i)是由方形及三角形函数内插的结果；(j)是作二维线性内插的结果。

在数字图像处理的实用系统中，用得最多的是双线性内插。例如，在处理过程中，有时要求计算在非整数的(k, l)坐标处的图像值 $f(k, l)$，其中 k 在 $m \sim m+1$ 之间，l 在 $n \sim n+1$ 之间。这时可以用双线性内插来实现上述计算，且 $f(k, l)$在计算中只和与其相邻的四个坐标位置上的图像取样值有关，如图 2.4 所示。

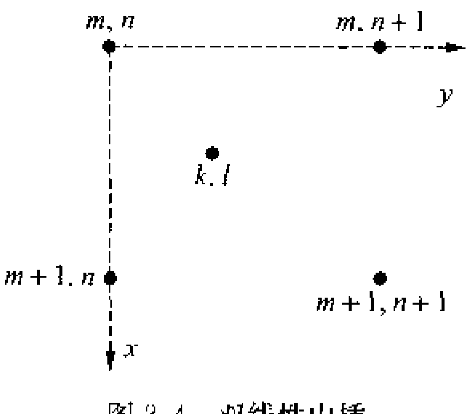


图 2.4　双线性内插

$f(k, l)$可由下式求出：

$$f(k, l) = (m+1-k)(n+1-l)f(m, n) + (k-m)(n+1-l)f(m+1, n) + (m+1-k)(l-n)f(m, n+1) + (k-m)(l-n)f(m+1, n+1) \quad (2.1-12)$$

从(2.1－12)式可看出，$f(k, l)$是四个角点上灰度值的加权平均，离(k, l)越近的点，其相应的权值就越大。

在一些数字图像处理系统中，重建可以用模拟低通滤波器来实现，即不必做内插计算，而只需将 $f(m, n)$经 D/A 转换后再用模拟低通滤波器进行滤波来得到模拟图像 $f(x, y)$。但是在很多情况下，还是需要用计算机进行内插计算。例如，在进行亚像素精度的运动估计运算时，或进行图像几何位置校正时，都必须用到内插计算。

2.1.3　截取和孔径[2]

前面关于取样的讨论是假设 $f(x, y)$在空间域无限延续，而实际上取样范围(取样窗)总是有限的，这就出现了“截取问题”。

从理论上分析，在严格意义上，不存在空(时)域和频域都有限的信号。例如正弦波在空域上为无限延续，但在频域上只是一个点。取有限的一段宽为 $M\Delta x$ 的信号，就是将 $f(x)$和取样窗函数 $\Pi\left(\frac{x}{M\Delta x}\right)$相乘，这相当于 $F(u)$和取样窗函数的频谱

(sin c 形)作卷积。即使原来 $F(u)$ 是限制在 $\pm W_u$ 频带范围内的，但经过卷积后就成为无穷宽，其限带性被破坏。但是如果“取样窗”很宽，即 $M\Delta x$ 很大，则其频谱拖尾处衰减很快，从实用角度看，问题不大。

在实际应用中，数字化的空间及频域处理要求对 $f(x, y)$ 和 $F(u, v)$ 都离散化，所以截取后空间图像相当于还要作周期延拓。如果截取时两端不一样大，则会有信号在周期延拓边缘处的“跳跃”现象。如果再作进一步处理(卷积)，则“边”上会模糊，这意味着这些处理结果无效。

相应的解决办法包括：加大取样窗、从截取后的原图做 DFT 时先补 0、用“窗口系数”相乘、丢弃“边缘”不准确的区域等。

取样孔径问题的由来是因为在从光→电→取样值的工程中，光路的“孔径”不会无穷小，取样脉冲宽度也不会无穷小，所以会产生很小的细节经取样后丢失的问题。例如，转鼓扫描把反射光在方形“孔径”上测得的光的强度作为孔径中心点处的取样值。如果“孔径”太小，不仅会带来因光能不够而产生的噪声影响，而且会在画面中有跳变时使取样得到的边缘变模糊。如果“孔径”太大，又会造成高频细节的丢失，使图像细节不清楚。因而在实际调试中，应选择适当大小的孔径。如果孔径太大，除去了高频成分，会造成图像的细节不清晰；如果孔径太小，光能不够，又会造成图像受噪声的影响太大。

2.2 量化

对图像 $f(x, y)$ 的取样完成之后，得到取样值 $f_s(m, n)$。在进入计算机前，$f_s(m, n)$ 还要进行量化。具体说，就是要在样本值的取值范围内进行分层(分成 k 个区间)，然后用单个值(逻辑值)来代表这一层内所有可能的值(物理值)。根据计算机内整数存放的惯例，一般可以把样本值的取值范围分成 $k=2^m$ 个层次，一般情况下 $m=6, 7, 8$，即可以把像素的灰度值分成 64，128 或 256 个层次(灰度级)。层次越多，则量化后的取样值(即计算机内的数字图像)恢复的实际图像越接近原图，看上去越令人满意。当层次不够时，在图像的灰度变换缓慢的区域内将出现原图上没有的假轮廓。这是由于量化过于粗糙，量化噪声过大在画面上的反映。这种假轮廓有可能掩盖了真实目标。图 2.5 中示出具有不同灰度量化层次(灰度级)的图像显示效果，在(f)和(g)中可看到明显的假轮廓。

在实际应用中，量化器的设计有三种不同的思路。第一种思路是使量化误差最小，即达到统计意义上的最优；第二种思路是根据人眼视觉特性来进行量化器的设计，例如对图像中人眼敏感的区域用细量化，而对人眼不敏感的区域用粗量化；



(a)　(b)　(c)

(d)　(e)　(f)

(g)　(h)

图2.5　不同灰度级的图像

(a)256×256，256灰度级图像；(b)～(h)为256×256，以128，64，32，16，8，4，2灰度级显示的图像

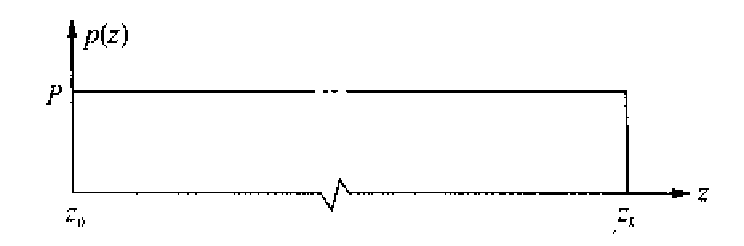
第三种是针对某些特定的应用、器件、设备或条件的量化策略。

最简单的量化方案是均匀量化，就是把样本值的整个取值范围均匀地分成k个子区间。然而，如果样本值在某个取值范围内频繁出现，而在其他范围内很少出现，按照第一种思路，就可以在样本值频繁取值的范围内采用小量化区间的密集量化，而在其他地方量化得粗糙一些。这样做可以在不增加量化层次的条件下，减少

实际引入的量化噪声,这就是非均匀量化。

2.2.1 均匀量化

取样完成后,得到 $z = f_s(m, n)$, $z \in [z_0, z_k)$。假设样本值的取值在$[z_0, z_k)$的范围内是同样频繁的,即概率密度 $p(z)$等于常数 P,则均匀量化是最佳量化。关于最佳量化的含义,将在下一小节中进行详细讨论。



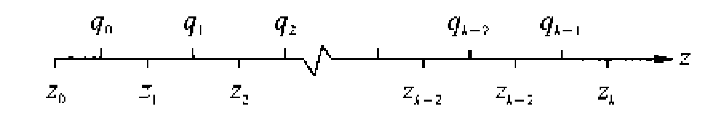


图 2.6 均匀量化图解

现在把整个取值范围$[z_0, z_k)$均匀地分成 k 个子区间$[z_i, z_{i+1})$, $i = 0, 1, \cdots, k-1$。如上所述,在数字图像中我们常取 $k = 64, 128, 256$。每个子区间$[z_i, z_{i+1})$由该子区间内一个确定的值q_i 来表示,总共由 k 个确定的值q_i 来代表所有可能出现的值,称为用 k 个层次进行量化。每一个确定的值 q_i,在计算机内由一个整数来表示。为方便起见,一般情况下这个整数与 q_i 成线性关系。计算机内部作处理时,仅关心这个整数的处理情况,并不关心相应的q_i 值。只有在恢复模拟信号及原图时,才关心 q_i 与这个整数的一一对应关系,这种关系通常由硬件加以保证。

假定已取样值 $z = f_s(m, n)$ 被量化成 $f(m, n)$,则

$$f(m, n) \in \{q_i\}, \ i = 0, 1, \cdots, k-1$$

量化过程很简单,即当

$$z = f_s(m, n) \in [z_i, z_{i+1})$$

则对应的量化值

$$f(m, n) = q_i \tag{2.2-1}$$

实际的量化器是 k 个电压比较器,各自的基准电压调到 $z_i (i = 0, 1, 2, \cdots, k)$。送入的 $f_s(m, n)$与这 k 个比较器比较,以确定 $f_s(m, n)$的值落入哪个子区间。这个值落入第 i 个子区间时,量化器直接送出一个与 q_i 相对应的整数值,作为 q_i 的代码。

在本书中，$f(m, n)$和q_i可以理解为一个灰度值，也可以理解为一个电平值、一个取样值（物理值）或一个编码值。读者可以由前后文加以判别。这里并没有具体标明用什么整数r_i来表示q_i，只是说明r_i和q_i一般呈线性关系，因为这仅仅是一个编码方案问题，并不是量化过程中的实质性问题。

现在来计算均匀量化引入的误差。当$z = f_s(m, n) \in [z_i, z_{i+1})$时，取样值量化为$f(m, n) = q_i$，这里引入了误差$(z - q_i)$。在$[z_i, z_{i+1})$区间取样值为$z$的概率密度为$p(z)$，因此在该区间内统计意义上引入的误差的平方为

$$\int_{z_i}^{z_{i+1}} (z - q_i)^2 p(z) \mathrm{d}z \tag{2.2-2}$$

前面已假设在$[z_0, z_k)$范围内概率密度$p(z)$等于常数P，因此k个子区间总的误差的平方ε^2为：

$$\begin{aligned}\varepsilon^2 &= \sum_{i=0}^{k-1} \int_{z_i}^{z_{i+1}} (z - q_i)^2 p(z) \mathrm{d}z = P \sum_{i=0}^{k-1} \int_{z_i}^{z_{i+1}} (z - q_i)^2 \mathrm{d}z \\ &= \frac{1}{3} P \sum_{i=0}^{k-1} [(z_{i+1} - q_i)^3 - (z_i - q_i)^3]\end{aligned} \tag{2.2-3}$$

当$\{q_i\}$确定之后，平均每个取样值量化引入的误差的平方ε^2也就确定了。

前面仅指出，$q_i \in [z_i, z_{i+1})$，并没有指出q_i在该区间中的具体位置，显然当q_i取值不同时，ε^2也不同。ε^2存在一个最小值。为取得ε^2的最小值，进行下面的运算：

取$\frac{\partial \varepsilon^2}{\partial q_i} = 0$，即

$$(z_{i+1} - q_i)^2 - (z_i - q_i)^2 = 0 \tag{2.2-4}$$

得到最佳量化值

$$q_i = \frac{1}{2}(z_{i+1} + z_i) \tag{2.2-5}$$

式(2.2-5)表明取样值$f_s(m, n)$在$[z_0, z_k)$内均匀分布，且量化值q_i取作每个子区间$[z_i, z_{i+1})$的中间值时，所引入的量化误差最小。

设子区间$[z_i, z_{i+1})$的长度为L，则式(2.2-3)中$P = \frac{1}{kL}$，这时

$$z_{i+1} - q_i = \frac{L}{2}, \; z_i - q_i = -\frac{L}{2}$$

上三式代入式(2.2-3)：

$$\varepsilon^2 = \frac{1}{3}\frac{1}{kL}\sum_{i=0}^{k-1}\left[\left(\frac{L}{2}\right)^3 - \left(-\frac{L}{2}\right)^3\right] = \frac{L^2}{12} \tag{2.2-6}$$

这就是每个取样值量化误差的平方值，这是一个在统计意义上的平均值。量化层次 k 加大时，L 成比例地缩小，则 ε^2 成平方反比地大大缩小。可见加大量化层次的数目 k 对保持原图的灰度信息十分有利。

2.2.2 非均匀量化

已有取样值 $z = f_s(m, n)$，其值分布在$[z_0, z_k)$之中，并已知在$[z_0, z_k)$中取值的概率密度为 $p(z)$。现在从引入的量化误差最小的角度来看应当怎样选定量化方式，即应当怎样确定每个子区间$[z_i, z_{i+1})$以及量化值 q_i。

q_i 是子区间$[z_i, z_{i+1})$中的一个确定值。子区间越大，引入的量化误差就越大。因此当概率密度 $p(z)$较小时，所取的量化区间长度 $z_{i+1}-z_i$ 可以大一些；反之，若在某一段区间上 $p(z)$较大时，则所选的量化区间 $z_{i+1}-z_i$ 就应小一些。用这种方法来达到统计意义上每个像素平均量化误差最小的目的。也就是说，当 $p(z)$不是常数时，量化区间的长度不一样，这就是非均匀量化。

设 $z = f_s(m, n)$，$z \in [z_0, z_k)$，要确定：

(1) 各个子区间$[z_i, z_{i+1})$。

(2) 各子区间对应的量化值 q_i。

式中：$i = 0, 1, \cdots, k-1$，即总共 k 个量化区间，以使得平方误差 ε^2 最小。ε^2 的表达式如下：

$$\varepsilon^2 = \sum_{i=0}^{k-1}\int_{z_i}^{z_{i+1}}(z-q_i)^2 p(z)\mathrm{d}z \tag{2.2-7}$$

现在求 ε^2 对 z_i 和 q_i 的偏导数，并使之为 0，即

$$\frac{\partial\varepsilon^2}{\partial z_i} = (z_{i-1}-q_i)^2 p(z) - (z_i - q_i)^2 p(z) = 0 \tag{2.2-8}$$

式中：$i = 1, 2, \cdots, k-1$，一共有 $k-1$ 个式子，这里 z_0 和 z_k 是已知的，不必对它们再求偏导数。另外令

$$\frac{\partial\varepsilon^2}{\partial q_i} = -2\int_{z_i}^{z_{i+1}}(z-q_i)p(z)\mathrm{d}z = 0 \tag{2.2-9}$$

式中：$i = 0, 1, \cdots, k-1$，一共有 k 个式子。

如果其中 $p(z) \neq 0$，由式(2.2-8)可得[2,13]

$$z_i = \frac{1}{2}(q_i + q_{i-1}),\ i = 1, 2, \cdots, k-1 \tag{2.2-10}$$

从式(2.2-9)可推导出：

$$q_i = \frac{\int_{z_i}^{z_{i+1}} zp(z)\mathrm{d}z}{\int_{z_i}^{z_{i+1}} p(z)\mathrm{d}z},\ i = 0, 1, \cdots, k-1 \tag{2.2-11}$$

由式(2.2-10)和(2.2-11)可知，最佳量化器的各子区间边界值 z_i 应当是量化值 q_i 间的中间值，而每一个 q_i 是子区间$[z_i, z_{i+1})$上由 $p(z)$构成的曲边梯形的形心。

由计算机的数值解法，可以求出由式(2.2-10)、(2.2-11)两者表达的 z_i 和 q_i。给定 z_0，z_k，$p(z)$，k 之后，求 z_i，q_i 的计算机解法如下：

(1) 先假定一个 q_0 的值，由(2.2-11)式取 $i = 0$，求出 z_1。由于 z_1 是积分式的上限，故要采用数值近似计算，逐步逼近 z_1 的解。

(2) 已知 z_1 和 q_0，在(2.2-10)式中取 $i = 1$，求出 q_1。

(3) 由 z_1 和 q_1，在(2.2-11)式中，取 $i = 1$，用数值近似计算逐步逼近 z_2 的解。

(4) 由 z_2 和 q_1，在(2.2-10)式中，取 $i = 2$，求出 q_2。

……

最终求出 q_{k-1}，代入(2.2-11)式，看其是否等于 $\frac{\int_{z_{k-1}}^{z_k} zp(z)\mathrm{d}z}{\int_{z_{k-1}}^{z_k} p(z)\mathrm{d}z}$。如果不相等，则重新假定 q_0 值，再回到第(1)步，继续做下去，求出 q_{k-1}，直到 q_{k-1} 符合要求为止[4]。这种方法由 Max 提出，可称为 Max 非均匀量化算法。

如果取样值在$[z_0, z_k)$范围内的所有地方同样频繁地出现，即 $p(z)$等于常数 P，式(2.2-10)和(2.2-11)可以简化为：

$$z_i = \frac{1}{2}(q_{i-1} + q_i),\ i = 1, 2, \cdots, k-1 \tag{2.2-12}$$

$$q_i = \frac{1}{2}(z_i + z_{i+1}),\ i = 0, 1, \cdots, k-1 \tag{2.2-13}$$

上两式表明这时非均匀量化已蜕化为均匀量化，见图 2.6。可见均匀量化是非均匀量化的一个特例。

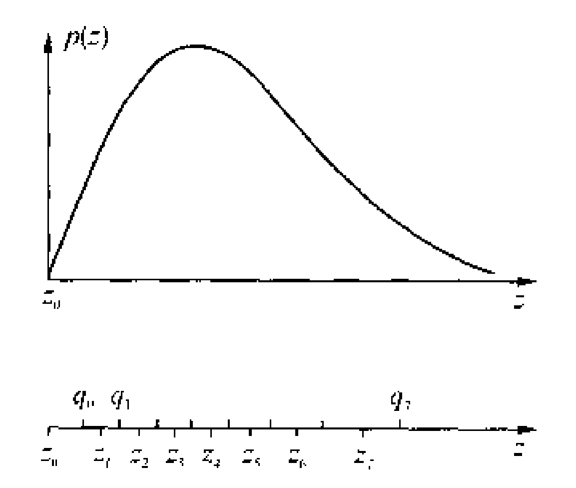
图 2.7(a)示出了一个非均匀量化的例子。在$[0, +\infty)$范围的各个子区间内，z 取值的频繁程度是不同的，即概率密度 $p(z)$不是常数，它满足所谓瑞利(Rayleigh)分布：

$$p(z)=\frac{z}{\sigma^2}\exp\left[-\frac{z^2}{2\sigma^2}\right] \tag{2.2-14}$$

图 2.7(a)中把$[0, +\infty)$划分成 $k=2^3=8$ 个量化层，对应着 8 个量化值 q_i。在 z 取值频繁的区段，即 $p(z)$较大的区段内，子区间$[z_i, z_{i+1})$较短；反之，子区间较长。同时也可注意到

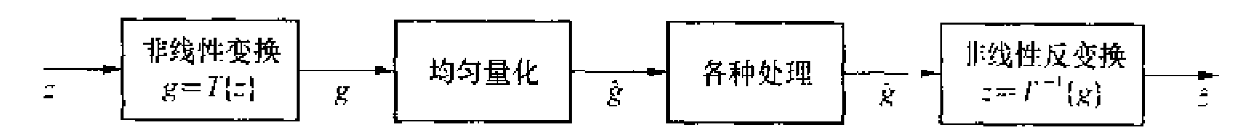
$$z_i=\frac{1}{2}(q_{i-1}+q_i)$$

其中列出的数据 $z_i(i=1, 2, \cdots, k-1)$ 和 $q_i(i=0, 1, \cdots, k-1)$ 是由计算机进行数值运算得出的解。



i	z_i	q_i
0	0.0000	0.5016
1	0.7619	1.0222
2	1.2594	1.4966
3	1.7327	1.9688
4	2.2182	2.4675
5	2.7476	3.0277
6	3.3707	3.7137
7	4.2124	4.7111


(a) 非均匀量化图解



(b) 压扩量化器示意图

图 2.7 非均匀量化

非均匀量化的概念和具体实施常见于数字通信和图像通信中。由图 2.7(b)所示的压缩扩展(Compand)运算可以实现非均匀量化。在这种量化器的压缩扩展系统中，先作一非线性变换[4][5]

$$g=T\{z\} \tag{2.2-15}$$

使变换后的值 g 的概率密度 $p'(g)$变为均匀的，然后进行 2.2.1 小节中所讨论的均匀量化。由于这时 $p'(g)$是均匀的，均匀量化也就是最佳量化。在进行了某些处理之后，再作一非线性反变换，即

$$z=T^{-1}\{g\} \tag{2.2-16}$$

以便重现图像。在图 2.7(a)所示的瑞利分布时，正变换和反变换表达式分别是

$$g = T\{z\} = \frac{1}{2} - \exp\left\{-\frac{z^2}{2\sigma^2}\right\} \tag{2.2-17}$$

$$z = T^{-1}\{g\} = \sqrt{2}\sigma\left[\ln\left\{\frac{1}{\frac{1}{2} - g}\right\}\right]^{\frac{1}{2}} \tag{2.2-18}$$

值得一提的是，式(2.2-17)是瑞利分布(2.2-14)式的累积分布函数，当 z 在 $[0, +\infty)$内分布时，g 在$[-0.5, -0.5)$内均匀分布，对 g 进行均匀量化是量化误差最小意义下的最佳量化。

到现在为止，进行非均匀量化时都是按第一种思路，即在统计意义上使量化误差最小的思路来设计量化器的。对于按人眼视觉特性进行量化器设计感兴趣的读者可参阅文献[8]。

在实现图像数字化的大多数系统中都采用均匀取样和量化的方案，即在空间沿 x 和 y 方向分别以等间隔 Δx 和 Δy 取样，取样点阵是正方形或长方形的。同时一律采用均匀量化不考虑灰度概率密度 $p(z)$的情况，这样做可使硬件及软件系统较简单，可适应大多数图像数字化的要求，但在一定程度上加大了量化噪声，并增大了数据的冗余度(即有部分数据多余)，这一点将在第 6 章图像数据压缩编码中详细讨论。

2.3　光度学基础知识

图像可以看成是一个二维的光辐射场。光可以从物理学中的辐射度学的角度进行度量，也可以从人眼对可见光的感觉，即光度学进行度量。光度学是光学中研究光在发射、传播、吸收、散射等过程中光的度量问题的学科。它结合人眼视觉特性研究各种光的度量的定义及其单位的选定。图像的摄取、数字化处理以后重新成像往往都要由人眼进行干预和判读，而且一些图像，如红外图像、X 光图像等也都要以可见光图像再现，以便由人眼判读和处理。因此了解和熟悉光度学中的基本量及其单位是很有必要的。

图 2.8 是光度学中对光的几个度量的示意图，它们是光源发光强度、光通量、照度和亮度。使用中往往把它们作为客观物理量来解释和利用，但事实上它们都是客观的光刺激人眼而引起的主观效果，包含了人眼视觉特性的影响。本节除了光的辐射功率是个客观物理量之外，所介绍的其他几种度量及其单位都与人眼视觉特性有关，而相对视敏函数曲线则表明了客观的光辐射功率与主观的人眼视觉特性之间的关系。

2.3.1 相对视敏函数

人眼对不同波长的光有不同的敏感度。波长不同而辐射功率相同的光不仅给人以不同色彩的感觉，而且亮度的感觉也不同。

人眼对波长约为555 nm的绿光在亮度上最敏感，而对红、紫色光则不敏感。经过对大量有正常视觉功能的人实验作出统计，得到了所谓相对视敏函数，如图2.9所示。

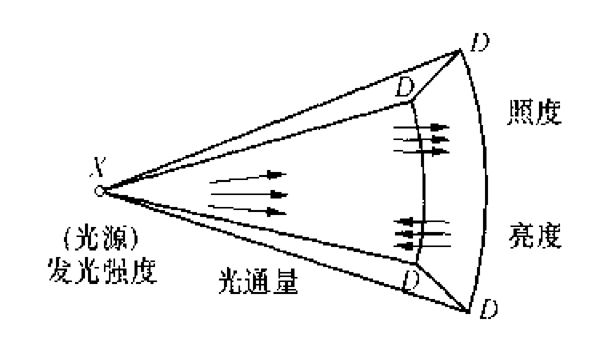


图2.8 光度学基本量示意图

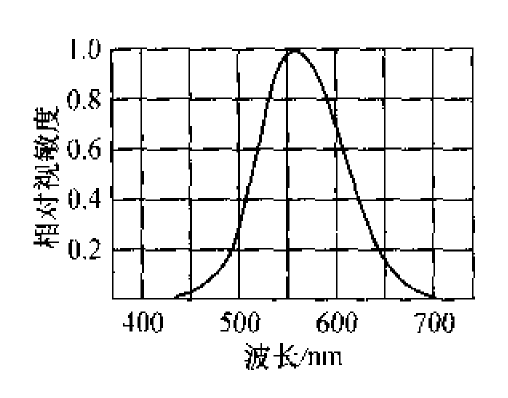


图2.9 相对视敏函数曲线

在产生相同亮度感觉的情况下，测出各种波长光的辐射功率 $P(\lambda)$。$P(\lambda)$越大，说明该波长的光越不容易被人眼所感觉；$P(\lambda)$越小，说明人眼对它的敏感程度越高。$P(\lambda)$的倒数可以用来衡量视觉对波长为 λ 的光的敏感程度，即视敏度，它用 $K(\lambda)$表示。即：

$$K(\lambda) = \frac{1}{P(\lambda)}$$

对波长 $\lambda = 555$ nm 的光，实验表明有最大的视敏度，即 $K_m = K(555)$。取任意波长的光的视敏度 $K(\lambda)$与 K_m 之比，就得到了相对视敏函数，用 $V(\lambda)$表示：

$$V(\lambda) = \frac{K(\lambda)}{K_m} = \frac{K(\lambda)}{K(555)} \qquad (2.3-1)$$

如果用得到相同主观亮度感觉时所需各种波长光的辐射功率 $P(\lambda)$来表示 $V(\lambda)$，则有

$$V(\lambda) = \frac{P(555)}{P(\lambda)} \qquad (2.3-2)$$

图2.9示出了相对视敏函数曲线，它是CIE(国际照明委员会)于1933年得到的，又称为相对光谱响应曲线。$V(\lambda)$的取值为 $0 \leqslant V(\lambda) \leqslant 1$。它说明为了得到相同的主观亮度感觉，在 $\lambda = 555$ nm 时，所需光的辐射功率为最小；波长 λ 偏离555 nm时，所需辐射功率将不断增大。当 $\lambda > 780$ nm或 $\lambda < 380$ nm 时，人眼对这些

辐射已没有亮度感觉,这些辐射已进入红外或紫外波段。

2.3.2 光通量

按人眼的光感觉来度量的辐射功率,称为光通量,记作 Φ。它的单位是 lm(流明)。利用光谱灵敏度曲线与人眼视敏曲线相同的光电管可以直接测量光通量。

由于人眼对不同波长光的亮度的敏感程度不同,因而以人眼的感觉来度量的某一波长光的辐射功率不仅与该波长光的辐射功率有关,而且也与人眼对该波长光的视敏度有关。

如果某单色光波长为 λ_1,光通量 $\Phi(\lambda_1)$就等于辐射功率 $P(\lambda_1)$与相对视敏函数 $V(\lambda_1)$的乘积:

$$\Phi(\lambda_1) = P(\lambda_1)V(\lambda_1) \tag{2.3-3}$$

国际上通用的光通量单位为流明。在 1 W 辐射功率下,555 nm 波长的单色光所产生的光通量为 680 lm。如果光源的辐射功率谱(辐射功率密度分布)为 $P(\lambda)$,则总的光通量为:

$$\Phi = 680\int_{380}^{780} P(\lambda)V(\lambda)\mathrm{d}\lambda(\mathrm{lm}) \tag{2.3-4}$$

功率相同的不同光源发出的光通量是不同的。例如,40 W 白炽灯的光通量为 468 lm,而 40 W 日光灯的光通量为 2 100 lm。

2.3.3 发光强度

光源在单位立体角内发出的光通量称为发光强度,简称光强,记作 I,单位为 cd(坎德拉)。它与光通量的关系是:

$$I = \frac{\mathrm{d}\Phi}{\mathrm{d}\omega}(\mathrm{cd}) \qquad \left(1\text{ 坎德拉} = \frac{1\text{ 流明}}{1\text{ 立体弧度}}\right) \tag{2.3-5}$$

在点光源的情况下,它向四周的光辐射是均匀的,而球心对球面的立体角为 4π 立体弧度,因而点光源的光强跟光通量的关系为:

$$I = \frac{\Phi}{4\pi},\ \Phi = 4\pi I \tag{2.3-6}$$

即 1 cd 发光强度的点光源,发出的总的光通量为 4π lm。

面光源的情况有所不同。多数面光源只在半球空间辐射,对于漫散射面(或称余弦散射面)光源,其光强是按余弦规律分布的,即与光源面法线成 α 角的光强

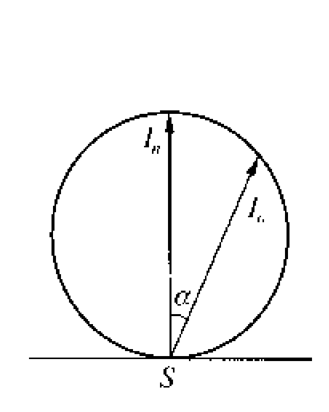
$$I_\alpha = I_0\cos\alpha \tag{2.3-7}$$

式中：I_n 为光源面法线方向的光强。光强分布示意图如图 2.10(a)所示。当 α 加大时，I_α 减小。

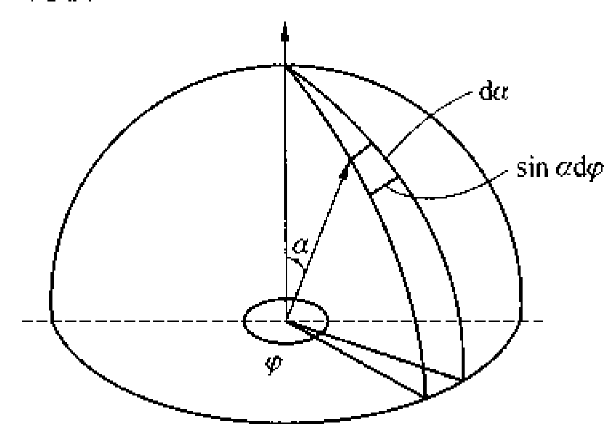
对于漫散射表面，在整个半球空间内所辐射的总的光通量

$$\begin{aligned}\Phi &= \int_0^{2\pi} \mathrm{d}\varphi \int_0^{\frac{\pi}{2}} I_\alpha \sin\alpha \mathrm{d}\alpha = \int_0^{2\pi} \mathrm{d}\varphi \int_0^{\frac{\pi}{2}} I_n \cos\alpha \sin\alpha \mathrm{d}\alpha \\ &= 2\pi I_n \times \frac{1}{2} \times \frac{1}{2} \cdot \cos 2\alpha \Big|_{\frac{\pi}{2}}^{0} = \pi I_n \end{aligned} \qquad (2.3-8)$$

读者可把(2.3-8)式与(2.3-6)式作一比较。



(a) 漫散射面光源的光强分布



(b) 半球空间内的总光通量计算

图 2.10 漫散射面光源

2.3.4 照度

光通量与被照射表面面积之比，称为照度，记作 E，单位为 lx(勒克斯)，即：

$$E = \frac{\mathrm{d}\Phi}{\mathrm{d}S}(\mathrm{lx}) \qquad \left(1\,\text{勒克斯} = \frac{1\,\text{流明}}{1\,\text{平方米}}\right) \qquad (2.3-9)$$

照度可由照度计直接测量。

被照表面与光源在空间的几何关系对照度有很大影响，下面以点光源为例说明。图 2.12 中点光源 A 向 dS 表面的引线与面积元 dS 的法线重合，引线长为 r。A 对 dS 构成立体角：

$$\mathrm{d}\omega = \frac{\mathrm{d}S}{r^2}$$

设点光源光强为 I，则由(2.3-5)求得被照面上的光通量为：

$$\mathrm{d}\Phi = I\mathrm{d}\omega = I\frac{\mathrm{d}S}{r^2} \qquad (2.3-10)$$

因此照度为：

$$E=\frac{\mathrm{d}\Phi}{\mathrm{d}S}=\frac{I}{r^2}=\frac{\Phi}{4\pi r^2} \tag{2.3-11}$$

当 A 向 $\mathrm{d}S$ 的引线与 $\mathrm{d}S$ 的法线间有一夹角 α 时，如图 2.11(b)所示，则：

$$\mathrm{d}\omega=\frac{\mathrm{d}S}{r^2}\cos\alpha$$

$$\mathrm{d}\Phi=I\mathrm{d}\omega=I\frac{\mathrm{d}S}{r^2}\cos\alpha \tag{2.3-12}$$

得到照度为：

$$E=\frac{I}{r^2}\cos\alpha=\frac{\Phi}{4\pi r^2}\cos\alpha \tag{2.3-13}$$

上式表明，被照表面的照度 E 与光源发光强度 I 成正比，与夹角 α 的余弦成正比，与光源到表面的距离 r 的平方成反比。当光源是平行光束时，照度将与距离无关。

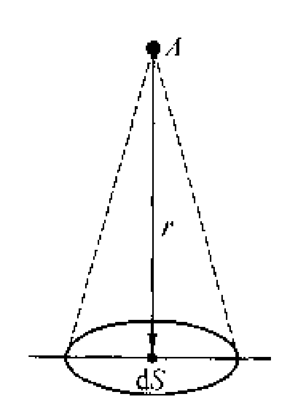
在室外，阴天的照度约 50～500 lx，而晴天可达 20 000 lx。

2.3.5　亮度

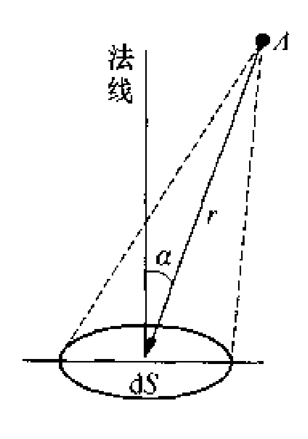
亮度表示发光面的明亮程度。它是指发光面在指定方向的发光强度与发光面在垂直于所取方向的平面上的投影之比，记作 B，即：

$$B=\frac{\mathrm{d}I_\alpha}{\mathrm{d}S\cos\alpha} \tag{2.3-14}$$

参见图 2.12，其中 $\mathrm{d}I_n$ 为发光面 $\mathrm{d}S$ 法线方向上的发光强度，$\mathrm{d}I_\alpha$ 为 $\mathrm{d}S$ 在指定方向上的发光强度，α 为该方向与发光面的法线方向之间的夹角。在取定方向上的发光强度 $\mathrm{d}I_\alpha$ 越大，在该方向上看到的发光面 $\mathrm{d}S\cos\alpha$ 越小，则看到的明亮程度越高，亮度 B 越大。在发光面的法线方向上：$B_n=\dfrac{\mathrm{d}I_n}{\mathrm{d}S}$。



(a) 点光源 A 与 $\mathrm{d}S$ 面垂直



(b) 点光源 A 与 $\mathrm{d}S$ 面不垂直

图 2.11　点光源下的照度

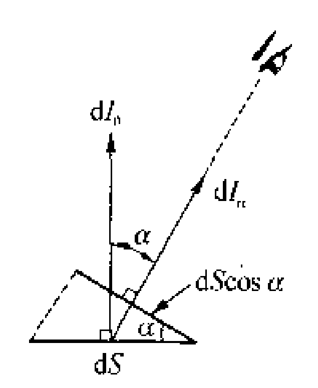


图 2.12　发光面的亮度

值得指出的是发光面可以是直接辐射的面光源，也可以是被光照射的反射面或透射面。

对于在 2.3.3 小节中讨论过的漫散射面，由(2.3-7)式可知，在 α 方向上的发光强度为

$$\mathrm{d}I_{\alpha} = \mathrm{d}I_{n}\cos\alpha,$$

由此在该方向上的亮度为：

$$B = \frac{\mathrm{d}I_{n}\cos\alpha}{\mathrm{d}S\cos\alpha} = \frac{\mathrm{d}I_{n}}{\mathrm{d}S} = B_{n} \tag{2.3-15}$$

上式说明，理想散射面虽然在各个方向上的光强和光通量都不同，但亮度却相同。

亮度可以用单位面积上的发光强度来表示，单位为 cd/m²(坎德拉每平方米)。

2.3.6 成像模型

许多图像是由物体将从光源来的光反射到眼睛或成像片上而形成的，如果物体受到均匀照射，即照度为常数 E，则图像的成像模型为

$$f(x, y) = Er(x, y) \tag{2.3-16}$$

其中 $r(x, y)$是反射成分，与物体的反射系数有关。在后面的章节中就使用这个模型，但不显式地表示出 E。在实际情况中，E 虽然不能达到严格意义上的均匀，但约等于均匀照明。

如果照度成分 $E(x, y)$随 x，y 变化而变化，即不均匀照明的情况，此时图像的成像模型为

$$f(x, y) = E(x, y)r(x, y) \tag{2.3-17}$$

由于上式中存在着乘性关系，用一般的滤波器难以去除光照不均匀的影响，因此发展出同态滤波技术，其处理过程将在第 4 章图像增强中介绍。

2.4 色度学基础知识

色度学是一门研究彩色计量的科学，其任务是研究人眼彩色视觉的定性和定量的规律及其应用。虽然由人眼来处理所觉察的彩色是一种生理和心理现象，对此至今还没有完全搞清楚，但借助于实验的和理论的结果能够表示彩色的物理性质。在数字图像的再现中，正确运用色度学原理，就能以简单而有效的技术手段再现真彩色、假彩色和伪彩色图像(详见第 4 章)。此外在制造彩色图像的硬拷贝过程中(如彩色印刷、

彩色胶片感光、彩色摄影),色度学原理的正确应用将有助于提高硬拷贝的质量。

彩色视觉是人的一种生理和心理现象。通过研究了解到,彩色视觉是人眼的一种明视功能,即在光照较明亮时,由视网膜中的锥状细胞起主要作用的视觉功能。

2.4.1 HSI 模型

就人眼视觉感受而言,各种颜色单从其波长不同来互相区分是不完全的,还应该同时采用色调(Hue)、色饱和度(Saturation)和亮度(Brightness)三个量。

色调反映颜色的类别,决定于彩色光的光谱成分,是彩色光在"质"方面的特征。如某物体在白光下呈现绿色,是因为它仅反射了绿色光分量。对于透射物体,其色调则由透射光的波长所决定。如果照射的不是白光,则人眼看到的颜色要由光源特性和物体反射/透射特性来共同决定。

色饱和度是某种彩色光纯度的反映,即彩色的浓淡,决定于彩色光中混入白光的数量。彩色光中纯光谱色的含量越多,其色饱和度越高。高饱和度的彩色光(如深红色)可以因掺入白光而被冲淡,变成低饱和度的彩色光(如淡红色)。

亮度决定于彩色光的强度,也可理解为彩色光引起视觉刺激的程度,是彩色光在"量"方面的特征。彩色光所含能量大则显得亮,反之则显得暗。

色调和色饱和度合称为"色度"。色度既说明了彩色光的颜色类别,又说明了颜色的深浅程度。

若用色环表示色调 H 和色饱和度 S,对应极坐标中的 θ 和 ρ,如图 2.13 所示;再用垂直轴表示亮度 I,则可用一个纺锤形颜色立体来表示彩色,如图 2.14 所示。

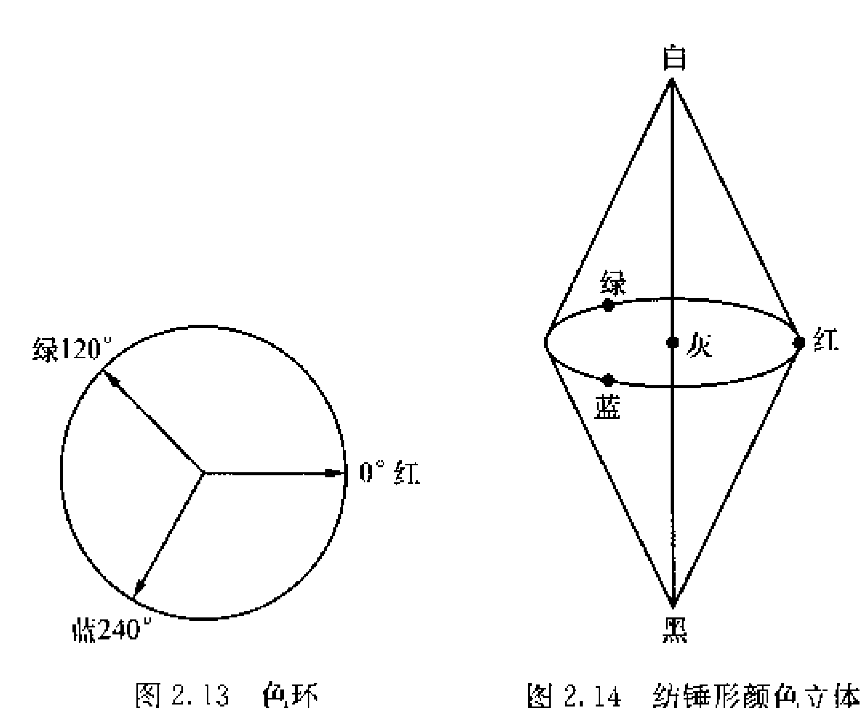


图 2.13 色环　　图 2.14 纺锤形颜色立体

从图 2.14 中可以看出,不同色调对应不同的角度 θ,例如红色对应 $\theta = 0°$。相

同色调、不同的色饱和度则表现为到"中心"的距离不同。越靠近中轴，色饱和度越趋近于0，彩色表现为中性色，如白色、灰色和黑色。在中轴上，越向上亮度越强，越向下越暗。

2.4.2 三基色原理

对于人眼的彩色视觉提出了这样一种假设：视网膜上的锥状细胞有三种类型：红敏细胞、绿敏细胞、蓝敏细胞。它们各自有自己的相对视敏函数曲线：$V_r(\lambda)$，$V_g(\lambda)$，$V_b(\lambda)$，如图2.15所示，三者相加得到图2.9所示的相对视敏函数曲线。

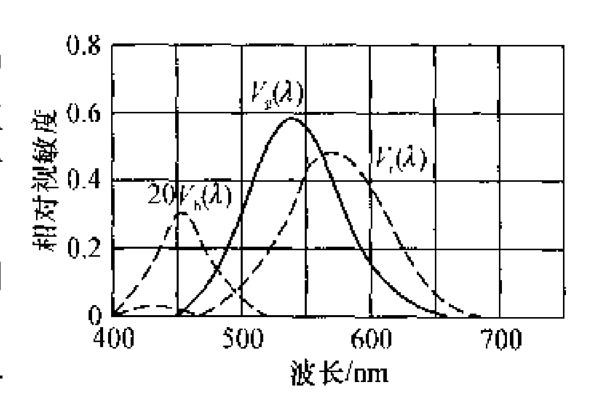


图2.15 三种锥状细胞的相对视敏函数曲线

当某一彩色光的功率谱为$P(\lambda)$时，对三种光敏细胞来讲，其光通量分别为：

$$\begin{cases} \Phi_r = \int_{380}^{780} P(\lambda)V_r(\lambda)\mathrm{d}\lambda \\ \Phi_g = \int_{380}^{780} P(\lambda)V_g(\lambda)\mathrm{d}\lambda \\ \Phi_b = \int_{380}^{780} P(\lambda)V_b(\lambda)\mathrm{d}\lambda \end{cases} \quad (2.4-1)$$

大脑根据Φ_r，Φ_g，Φ_b的比例判定总的色度(H，S)，三者合成的总光通量决定了总的亮度(I)。若有两束光谱成分不同的光，但只要对应的Φ_r，Φ_g，Φ_b相同，人眼主观上的彩色感觉(色度和亮度)就相同。三种锥状细胞的假说虽然符合实验结果，但还需从生理解剖上得更充分的证实。

不同波长的单色光会引起不同的彩色感觉，然而，同样的彩色感觉却可以来源于不同的光谱成分的组合。这就是说，光谱分布跟彩色感觉之间的关系是多对一的。例如呈现白光的日光是一种连续光谱，但也可由红、绿、蓝三种单色光以适当比例混合而成。为得到相同的彩色感觉，不必还原"原景物"的光谱成分，可以由不同光谱的光组合起来，来得到相同的彩色感觉。于是可以选择三种基色，把它们按不同比例组合，便可以引起不同的彩色感觉。对人眼彩色感觉来说，如果这三种基色选择合适，几乎自然界中的所有彩色都可由这三种基本彩色光混合而成，这就是三基色原理。

在彩色电视及数字图像处理系统的彩色监视器上，采用在红色(R)、绿色(G)和蓝色(B)光谱区域内选择的三个基色，把它们按不同比例相加而获得不同的彩色。这种方法称为相加混色法，可以由下面几种方法实现彩色的相加：

(1) 把三种光同时投射到一个全反射的表面上合成。

(2) 把三种光轮流投射到一个全反射的表面上。由于视觉惰性，只要轮换的速度足够快，人眼彩色感觉跟同时投射的情况相同。这个方法称为时间混合法。

(3) 同时把三种光分别投射到同一表面上相邻的三个点上，只要这些点相距足够近，有限的视觉分辨率可以产生三种光相混合的彩色感觉。这个方法称为空间混合法。

(4) 两只眼睛同时分别看两种不同颜色的同一图像，产生混色效果。这个方法称为生理混色法。

若使用上面第一种方法，将 R, G, B 三个"色斑"投到白屏上，调节三色的亮度，使中间区域为白色，则产生的混色效果如图 2.16 所示。

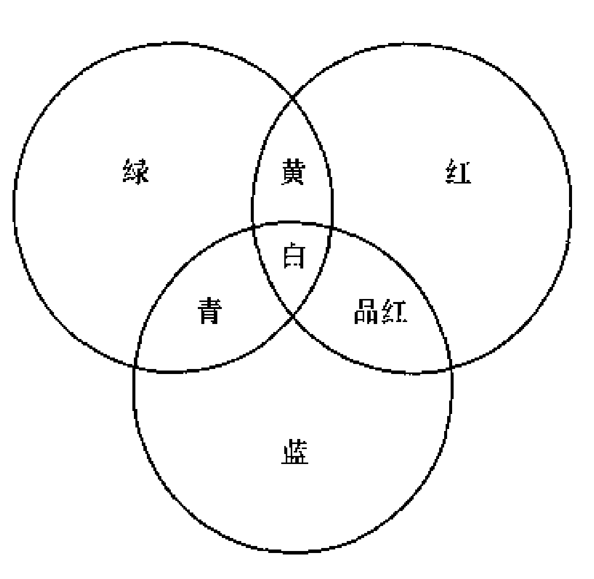


图 2.16　相加混色

从图 2.16 可看出：

红＋绿＝黄
蓝＋绿＝青
红＋蓝＝品红
红＋绿＋蓝＝白

换一个角度看问题，有下列等式成立：

黄＝白－蓝　即：黄是蓝的补色
青＝白－红　即：青是红的补色
品红＝白－绿　即：品红是绿的补色

由上式可知，两个互补的色相加等于白。

在彩色印刷、彩色胶片感光中采用的是相减混色法。它是利用颜料、染料的吸色性质实现的。在相减混色法中常采用黄色、品红、青色为三基色。它们可以分别吸收各自的补色光，即蓝光、绿光、红光。把这三种基色按不同比例相混合后，在白光照射下，蓝光、绿光和红光以相应比例被吸收，便呈现不同的彩色。具体实现如下：

白光投在黄色颜料上，颜料吸收了蓝光，所以呈黄色，即

白－蓝＝黄

白光投在青色颜料上，颜料吸收了红光，所以呈青色，即

白－红＝青

把黄和青两种颜料混合，则蓝光和红光都被颜料吸收，只反射绿光，所以呈绿

色，即

$$黄+青=白-蓝-红=绿$$

同样，

$$黄+青+品红=白-蓝-红-绿=黑。$$

上述关系可用图 2.17 表示。

对于相加混色和相减混色，可以用彩色立方体来统一表示，如图 2.18 所示。

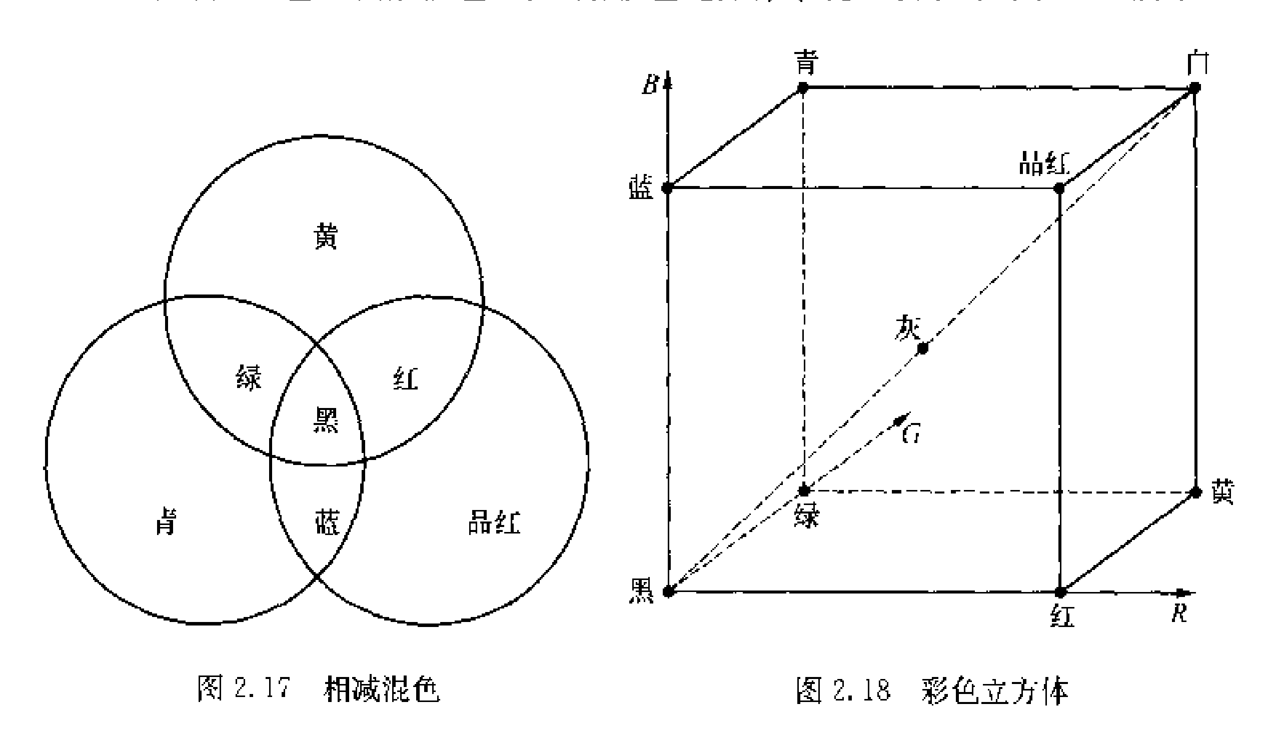


图 2.17 相减混色　　图 2.18 彩色立方体

从图 2.18 可以看出：加色坐标系是以 R，G，B 为三个坐标轴构成的坐标系；坐标轴上三个顶点分别对应 R，G，B 三基色；原点上没有 R，G，B 光，表现为黑色；对角点上 R，G，B 光都达到最亮，表现为白色；在对角线附近，R，G，B 分量相等，表现为灰色；另外三个顶点分别对应三个补色，即青、黄、品红；以白色为顶点，以青、黄、品红为三个坐标轴即构成了减色坐标系。

2.4.3 物理三基色和 RGB 三色系数

选择三基色的原则是：获得方法简单、色度稳定而准确、配出的彩色尽可能多。为此，CIE（国际照明委员会）规定了表 2.1 的三种光为基色，它们是水银光谱中的三种谱色光，并确定了基色单位当量。当量是根据配出标准白光 $E_{白}$（又称等能白光，即在可见光谱范围内所有波长的光与色温 5 500 K 的白光相近）时，三基色的光通量比例来确定的。

$$\Phi_r : \Phi_g : \Phi_b = 1 : 4.5907 : 0.0601 \qquad (2.4-2)$$

表 2.1 *RGB* 物理三基色

基色	波长	基色单位当量	基本单位符号
红	700.0 nm	1.0000	$[R]$
绿	546.1 nm	4.5907	$[G]$
蓝	435.8 nm	0.0601	$[B]$

在确定了基色单位之后，白光 $E_{白}$ 的配色关系表示为：

$$F_{E_{白}} = 1[R] + 1[G] + 1[B] \tag{2.4-3}$$

式中：Φ 表示光通量；F 表示具有一定亮度和色度的彩色光。

对于任意给定的彩色光 F，其配色方程可写成

$$F = R[R] + G[G] + B[B] \tag{2.4-4}$$

对应光通量

$$\Phi = 680(R + 4.5907G + 0.0601B)(\text{lm}) \tag{2.4-5}$$

式中：R, G, B 称为三色系数（它们可以由配色实验测出）。R, G, B 的比例关系决定了所配彩色光的色度（色调和色饱和度）；它们的数值决定了所配彩色光的亮度。$R[R]$, $G[G]$, $B[B]$称为 F 色光的三色分量。

表 2.1 所选择的三基色，其物理意义清楚，称为物理三基色。

2.4.4 计算三基色和 *XYZ* 三色系数

上一小节所讨论的物理三基色在使用上不方便，首先某一彩色光的光通量（亮度与之成正比）要另外按（2.4-5）式计算，其次在配置高饱和的绿色或蓝色光时，三色系数 R, G, B 会出现负值等等。为克服这些缺点，CIE 另外规定了计算三基色，其基色单位为$[X]$, $[Y]$, $[Z]$。它们并不代表真实的彩色，但在计算上可以克服物理三基色的缺点，并可产生实用的 *XYZ* 色度图（CIE 色度图）。

虽然$[X]$, $[Y]$, $[Z]$是一组虚基色，但在数学上它们可以由物理三基色以线性组合的方式配出。计算三基色的单位$[X]$, $[Y]$, $[Z]$与物理三基色$[R]$, $[G]$, $[B]$之间的关系可以用矩阵表示为：

$$\begin{bmatrix}[X]\\ [Y]\\ [Z]\end{bmatrix} = [\mathbf{A}]\begin{bmatrix}[R]\\ [G]\\ [B]\end{bmatrix} \qquad [\mathbf{A}] = \begin{bmatrix}0.4185 & -0.0912 & 0.0009\\ -0.1587 & 0.2524 & -0.0025\\ -0.0828 & 0.0157 & 0.1786\end{bmatrix} \tag{2.4-6}$$

$$\begin{bmatrix}[R]\\ [G]\\ [B]\end{bmatrix}=[\boldsymbol{A}]^{-1}\begin{bmatrix}[X]\\ [Y]\\ [Z]\end{bmatrix}\qquad [\boldsymbol{A}]^{-1}=\begin{bmatrix}2.7689 & 1.0000 & 0.0000\\ 1.7518 & 4.5907 & 0.0565\\ 1.1302 & 0.0601 & 5.5943\end{bmatrix}\tag{2.4-7}$$

由式(2.4-6)和式(2.4-7)可以求出在 RGB 三色系数和 XYZ 三色系数之间的关系。已知某彩色光 F 在 RGB 基色下的配色方程，然后代入式(2.4-7)，有：

$$F=R[R]+G[G]+B[B]=[R,\ G,\ B]\begin{bmatrix}[R]\\ [G]\\ [B]\end{bmatrix}$$

$$=[R,\ G,\ B][\boldsymbol{A}]^{-1}\begin{bmatrix}[X]\\ [Y]\\ [Z]\end{bmatrix}=[X,\ Y,\ Z]\begin{bmatrix}[X]\\ [Y]\\ [Z]\end{bmatrix}$$

推出 F 在 XYZ 基色下的三色系数为

$$[X,\ Y,\ Z]=[R,\ G,\ B][\boldsymbol{A}]^{-1}$$

或写成：

$$\begin{bmatrix}X\\ Y\\ Z\end{bmatrix}=([\boldsymbol{A}]^{-1})^{\mathrm{T}}\begin{bmatrix}R\\ G\\ B\end{bmatrix}\qquad ([\boldsymbol{A}]^{-1})^{\mathrm{T}}=\begin{bmatrix}2.7689 & 1.7518 & 1.1302\\ 1.0000 & 4.5907 & 0.0601\\ 0.0000 & 0.0565 & 5.5943\end{bmatrix}\tag{2.4-8}$$

同理可有：

$$F=X[X]+Y[Y]+Z[Z]=[X,\ Y,\ Z]\begin{bmatrix}[X]\\ [Y]\\ [Z]\end{bmatrix}$$

$$=[X,\ Y,\ Z][\boldsymbol{A}]\begin{bmatrix}[R]\\ [G]\\ [B]\end{bmatrix}=[R,\ G,\ B]\begin{bmatrix}[R]\\ [G]\\ [B]\end{bmatrix}$$

得到：

$$\begin{bmatrix}R\\ G\\ B\end{bmatrix}=[\boldsymbol{A}]^{\mathrm{T}}\begin{bmatrix}X\\ Y\\ Z\end{bmatrix}\qquad [\boldsymbol{A}]^{\mathrm{T}}=\begin{bmatrix}0.4185 & -0.1587 & -0.0828\\ -0.0912 & 0.2524 & 0.0157\\ 0.0009 & -0.0025 & 0.1786\end{bmatrix}\tag{2.4-9}$$

式(2.4-8)和(2.4-9)给出了两组基色下的三色系数之间的关系。

2.4.5 XYZ 色度图(CIE 色度图)

彩色光 F 的表达式在其三色系数归一化后可以写成：

$$F = X[X] + Y[Y] + Z[Z] = m'(x[X] + y[Y] + z[Z]) \quad (2.4-10)$$

式中：

$$\begin{aligned} m' &= X + Y + Z \\ x &= \frac{X}{X+Y+Z} = \frac{X}{m'} \\ y &= \frac{Y}{X+Y+Z} = \frac{Y}{m'} \\ z &= \frac{Z}{X+Y+Z} = \frac{Z}{m'} \end{aligned} \quad (2.4-11)$$

因此有

$$x + y + z = 1 \quad (2.4-12)$$

x，y，z 称为 XYZ 制下的色度坐标或相对色系数。在式(2.4-12)中三个色度坐标中有一个是不独立的，因此可以用 $x-y$ 直角坐标来表示各种色度，这样的平面图形就是 XYZ 色度图(即 CIE 色度图)，见图 2.19。

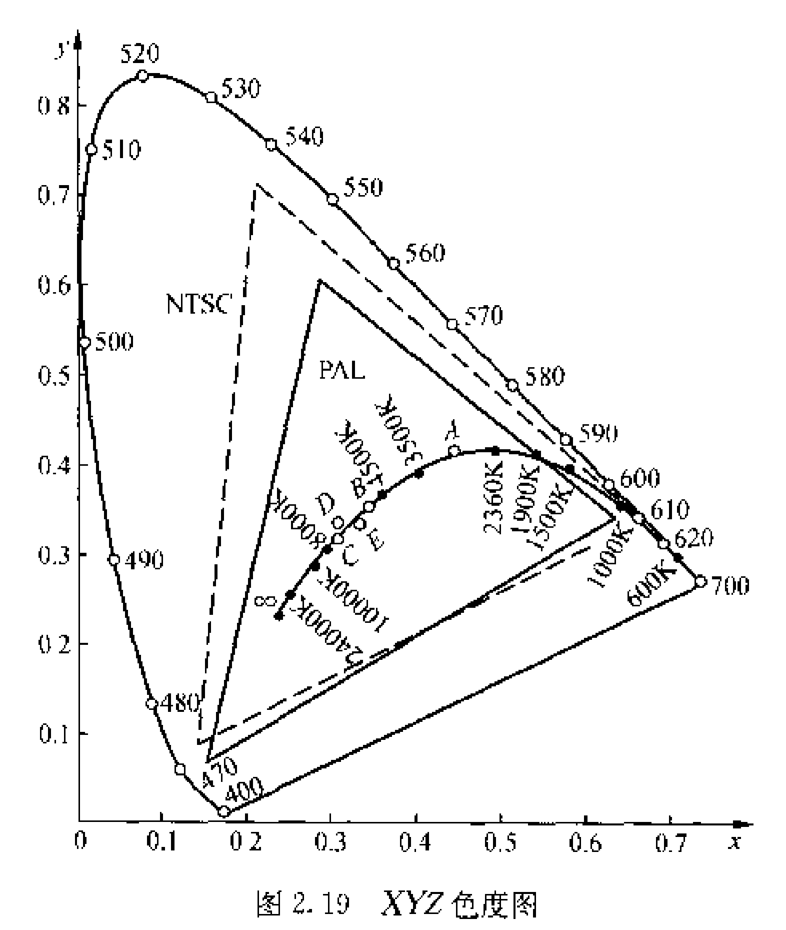


图 2.19 XYZ 色度图

从图 2.19 可见，所有的谱色光都位于舌形线上，该线又称谱色轨迹。在舌形线上加注了对应的波长数。谱色轨迹不包含舌形线的底部，该直线上的颜色是由红基色和蓝基色混合得到的非谱色光，不能作为谱色光出现在光谱上。

自然界中的彩色都能用整个闭合曲线及其内部相应点的坐标表达，并可由物理方法实现。坐标位置越靠近谱色轨迹，所对应的彩色越纯；越靠近 E 点（称 $E_白$），所对应的彩色的饱和度越低。

与 RGB 制相比，XYZ 制及其色度图（CIE 图）有下述优点：

(1) 在作色度计算及配制彩色时，各 X，Y，Z 系数均为正值。

(2) 对比式(2.4-5)及(2.4-8)中矩阵$(\boldsymbol{A}^{-1})^T$ 的第二行可知，由 Y 可确定彩色光的亮度，并规定 1[Y]的光通量为 680 lm。

(3) 彩色光的色度仍然由 X、Y、Z 的比值确定。

(4) 当 $X = Y = Z$ 时，对应 $R = G = B$，代表了等能白光 $E_白$。

通常用 x 和 y 表示色度，用 Y 表示亮度。若已知两束彩色光分别为 $F_1(x_1, y_1, Y_1)$和 $F_2(x_2, y_2, Y_2)$，则合成光 F_{1+2}的参数为：

$$Y_{1+2} = Y_1 + Y_2 \tag{2.4-13}$$

$$\begin{cases} x_{1+2} = \dfrac{m_1' x_1 + m_2' x_2}{m_1' + m_2'} \\ y_{1+2} = \dfrac{m_1' y_1 + m_2' y_2}{m_1' + m_2'} \end{cases} \quad 其中 \begin{cases} m_1' = \dfrac{Y_1}{y_1} \\ m_2' = \dfrac{Y_2}{y_2} \end{cases} \tag{2.4-14}$$

现在对上两式作一简单证明。设有两束彩色光：

$$F_1 = m_1' \{x_1[X] + y_1[Y] + z_1[Z]\}$$
$$F_2 = m_2' \{x_2[X] + y_2[Y] + z_2[Z]\} \tag{2.4-15}$$

配色实验表明，把 F_1 和 F_2 相混作为待配彩色光 F_{1+2}，则其各基色分量的系数正好等于为了分别配出 F_1 和 F_2 所需该基色分量的系数之和：

$$\begin{aligned} F_{1+2} &= F_1 + F_2 = (m_1' x_1 + m_2' x_2)[X] + (m_1' y_1 + m_2' y_2)[Y] + (m_1' z_1 + m_2' z_2)[Z] \\ &= (m_1' + m_2')\left\{\frac{m_1' x_1 + m_2' x_2}{m_1' + m_2'}[X] + \frac{m_1' y_1 + m_2' y_2}{m_1' + m_2'}[Y] + \frac{m_1' z_1 + m_2' z_2}{m_1' + m_2'}[Z]\right\} \\ &= m_{1+2}' \{x_{1+2}[X] + y_{1+2}[Y] + z_{1+2}[Z]\} \end{aligned} \tag{2.4-16}$$

由上式可以得到式(2.4-14)。而且

$$Y_{1+2} = m_{1+2}' y_{1+2} = m_1' y_1 + m_2' y_2 = Y_1 + Y_2 \tag{2.4-17}$$

上述计算也可在CIE色度图上用图解法完成，如图2.20所示。

设F_1，F_2，F_{1+2}分别位于$A(x_1, y_1)$，$B(x_2, y_2)$和$C(x_{1+2}, y_{1+2})$。则由式(2.4-14)很容易证明A，B，C三点共线。设AC，CB，AB长度为L_1，L_2和L，不难证明：

$$L_1 = \frac{m_2'}{m_1' + m_2'}L, \qquad L_2 = \frac{m_1'}{m_1' + m_2'}L \tag{2.4-18}$$

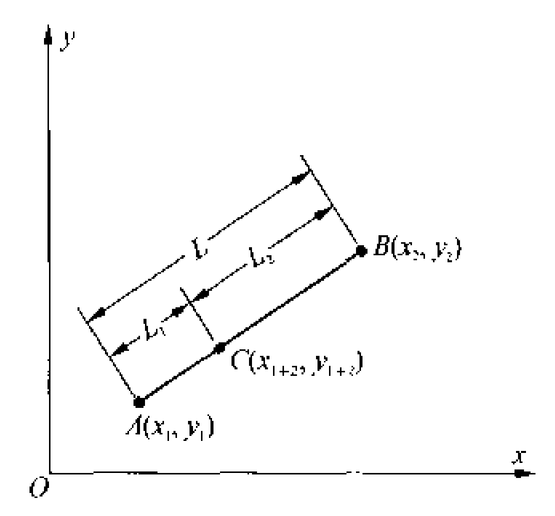


图2.20 色度图解计算

得到：

$$L_1 : L_2 = m_2' : m_1', \qquad L_1 + L_2 = L \tag{2.4-19}$$

因此当A和B的“重量”为m_1'和m_2'时，C是它们的“重心”。

不同的彩色监视器所选取的显像三基色有所不同，即它们的色度坐标不同。表2.2给出了在NTSC和PAL彩色电视制式中所采用的显像三基色在XYZ色度图中的色度坐标以及它们的标准白光$C_白$和D_{65}的色度坐标。

图2.19中示出表2.2所列显像三基色坐标点的位置及由它们构成的三角形。三角形面积代表了由这些基色所能配出的彩色范围。

表2.2 显像三基色的色度坐标

制式		PAL制				NTSC制			
基色与光源		$[R_e]$	$[G_e]$	$[B_e]$	D_{65}	$[R_e]$	$[G_e]$	$[B_e]$	$C_白$
色坐标	x	0.64	0.29	0.15	0.313	0.67	0.21	0.14	0.310
	y	0.33	0.60	0.06	0.329	0.33	0.71	0.08	0.316

2.4.6 一些常用公式

在CCIR601标准(ITU-R.601)和SMPTE 125M标准中都用Y，C_b，C_r来表示彩色图像。其中Y为亮度，C_b，C_r为色差，且有：

$$C_b = \frac{B - Y}{2} + 0.5$$
$$C_r = \frac{R - Y}{1.6} + 0.5 \tag{2.4-20}$$

对NTSC制(平衡正交调制)的彩色电视，其亮度公式为：

$$Y = 0.2999R + 0.587G + 0.114B \qquad (2.4-21)$$

对 PAL 制(逐行倒相制)的彩色电视,其亮度公式为:

$$Y = 0.222R + 0.707G + 0.071B \qquad (2.4-22)$$

有时用 u, v 来表示色差,u, v 与 C_b, C_r 只相差系数和偏移量,其对应关系如下:

$$\begin{cases} u = B - Y \\ v = R - Y \end{cases} \qquad \begin{cases} C_b = \dfrac{u}{2} + 0.5 \\ C_r = \dfrac{v}{0.6} + 0.5 \end{cases} \qquad (2.4-23)$$

2.5 人眼视觉特性

人类对人眼视觉特性进行了大量的实验和研究。虽然目前在机理上还未完全弄清楚这些特性,但黑白及彩色电视的实用化及许多图像工程的成功,表明对人眼视觉特性的了解是经受了考验的。本节仅对跟数字图像处理关系密切的一些特性作一番介绍。

2.5.1 人眼构造简介

人眼的构造和照相机类似,见图 2.21(a)。它的正前方 1/6 部分是透明的角膜。角膜后面是不透明的虹膜,虹膜中间有一圆孔称为瞳孔,在虹膜环状肌组织的帮助下,瞳孔的直径可以在 2～8 mm 间调节,从而控制进入人眼内部的光通量,起到照相机中光圈的作用。瞳孔后面是一扁球形弹性透明体,称为晶状体,相当于照相机中的透镜。在睫状肌的作用下,晶状体的曲率可以调节以改变焦距,使不同距离的景物可以在视网膜上成像。

在正对晶状体轴线的视网膜上有一个集中了大量锥状细胞的黄斑区(又称视网膜的中央凹)。锥状细胞的直径为2～6μm,长约40μm,共达 500 万至 700 万个。它们主要是在强光(明视)之下检测亮度和颜色信息。每个锥状细胞都连接着一个神经末梢,因此黄斑区的分辨率很高。在视网膜的其他部分分布着杆状细胞,其直径为2～4μm,长约60μm,共有7 500万至15 000万个。它们能在弱光(暗视)下检测亮度信息,但没有色彩的感觉。多个杆状细胞跟同一跟神经末梢相连,故对细节的分辨率较低,只能给出视野中一般的总轮廓。依靠眼球的转动使目标落在黄斑区上成像以便看清物体。

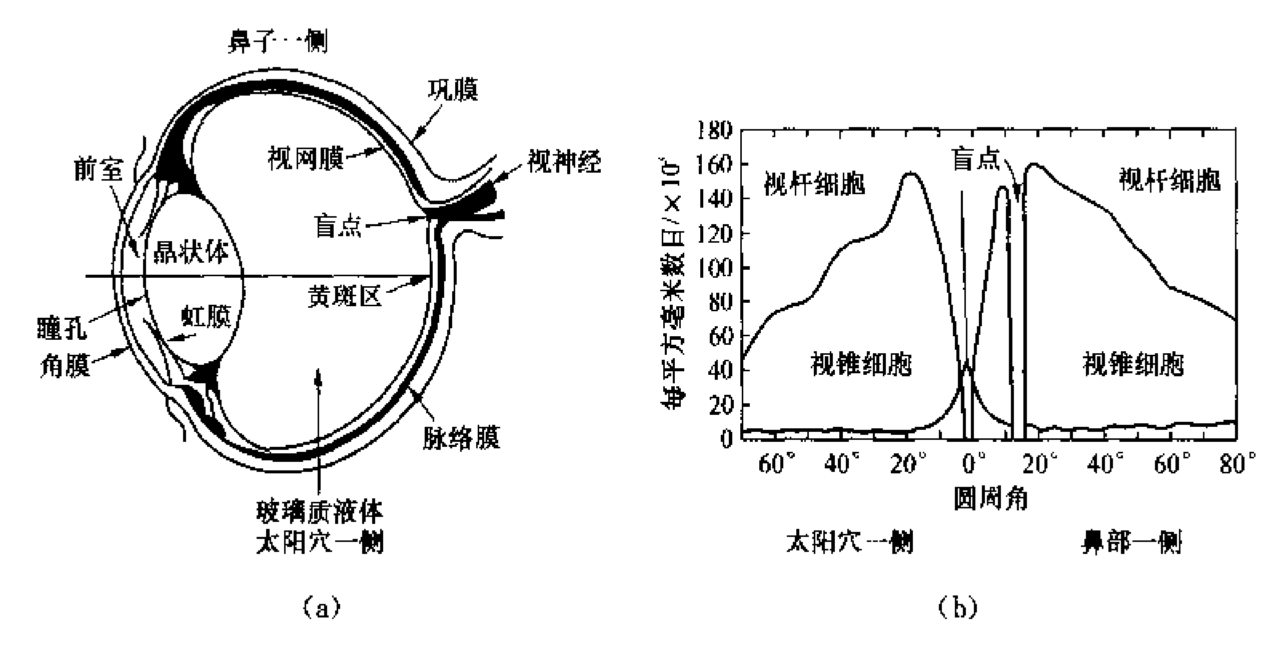


图 2.21 人眼的结构与视细胞的分布

锥状细胞和杆状细胞统称视细胞。图 2.21(b)示出了视细胞在视网膜上分布的情况,以眼球水平方向上的圆周角作为水平距离的度量,黄斑中心为 0°。在向鼻侧 20°的地方是视神经的汇集点,没有视细胞,成为"盲点"。对于落入盲点中的这部分图像入眼将无法看到。

视细胞受到光的刺激产生电脉冲。电脉冲沿着神经纤维传递到视神经中枢。由于各细胞产生的电脉冲不同,大脑就形成了景象的感觉。

2.5.2 明暗视觉及视觉范围

图 2.9 是白天正常光照下人眼对不同波长光在亮度方面敏感程度的曲线,又称明视觉相对视敏函数曲线,其峰值在555 nm处。明视觉过程主要由锥状细胞完成。这时除了有明暗感觉外,还有彩色感觉。在微弱光线下,整个曲线左移,峰值移到 505 nm 附近,这时紫色能见范围扩大而红色能见范围缩小。该曲线称为暗视觉相对视敏函数曲线,如图 2.22 中虚线所示。暗到一定程度,仅有杆状细胞起作用,这时因分辨不出颜色,

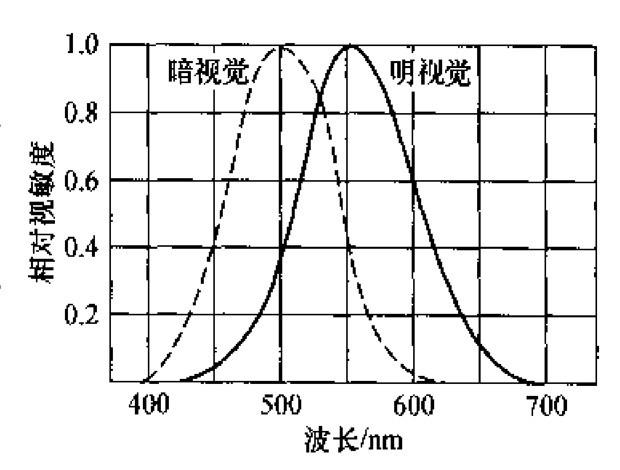


图 2.22 明暗视觉的视敏函数曲线

整个光谱带看上去成为明暗不同的灰度带。

人眼所能感觉到的亮度范围称为视觉范围。它很宽，可以从 10^{-4} 至 10^{6} cd/m²。如图 2.23 所示。人眼是通过随外来光强弱的自动调节来适应这样宽的亮度范围的。这种调节主要依靠视细胞本身的调节作用，其次是瞳孔的调节作用。

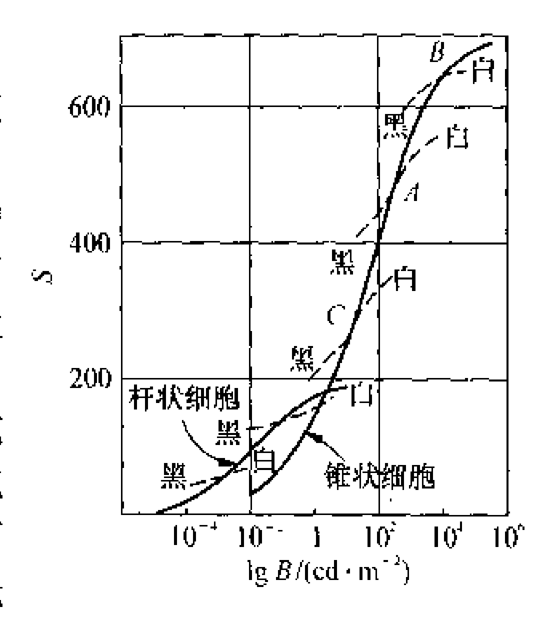


图 2.23 眼睛的亮度感觉

人眼并不能同时感受这样大的亮度范围。当人眼适应了某一环境亮度后，视觉范围就有了一定的限度。在适当的平均亮度下能分辨亮度的上下限之比为1 000∶1；在很低的平均亮度下，这一比值为 10∶1。

人眼对明暗程度所形成的“黑”、“白”感觉具有相对性，还表现在不同环境亮度下对同样亮度的主观感觉并不相同。

2.5.3 亮度感觉

数字图像用黑白方式再现时是作为许多分离的亮点显示出来的。因此人眼对于不同亮度之间的鉴别能力是图像处理中所要考虑的一个重要方面。在 2.3 节中定义亮度时已经考虑了人眼的相对视敏函数曲线。但是在实际观察景物时得到的亮度感觉并不完全由景物的亮度决定。下面用三个实验的结果来说明亮度感觉的一些规律。

让我们观察一个面积很大，亮度为 B 的发光面。现在不断变化 B，人眼主观上感觉到亮度在变化，估出主观亮度 S。以 $\lg B$ 为横坐标、主观亮度 S 为纵坐标得到主观亮度与亮度关系的 $\lg B-S$ 曲线，如图 2.23 所示。可以看到该曲线较长的一段是线性的，但两端弯曲。曲线下部有相交的两个分支，表明两种视细胞的感觉不同，反映了暗视觉和明视觉是逐渐过渡的。在实验中应使亮度变化足够慢，使人眼的适应过程达到稳定。

当亮度强弱不同时，人眼的适应能力使得从主观上难以正确判定亮度的绝对大小。但在比较两个亮度时，人眼有较好的判断能力。为此做第二个实验，观察一个面积较大亮度为 B 的背景及其上一个亮度为 $B+\Delta B$ 的边界鲜明的目标，如图 2.24(a)所示。开始 ΔB 为 0，逐步加大 ΔB，直至 B 与 $B+\Delta B$ 可以分辨为止，记这时 $\Delta B=\Delta B_{\min}$。在相当大的亮度范围内，可觉察的最小相对亮度变化 $\frac{\Delta B_{\min}}{B}$ 为一常数 ξ，这个关系称为韦伯定律，ξ 称为韦伯比。在很宽范围内 ξ 是 0.02，亮度很高

或很低时，ξ可以迅速上升。把实验结果以

$$\lg B—\frac{\Delta B_{min}}{B}$$

曲线画出，如图2.24(b)所示。曲线两端上升表明韦伯比上升，亮度分辨率下降。

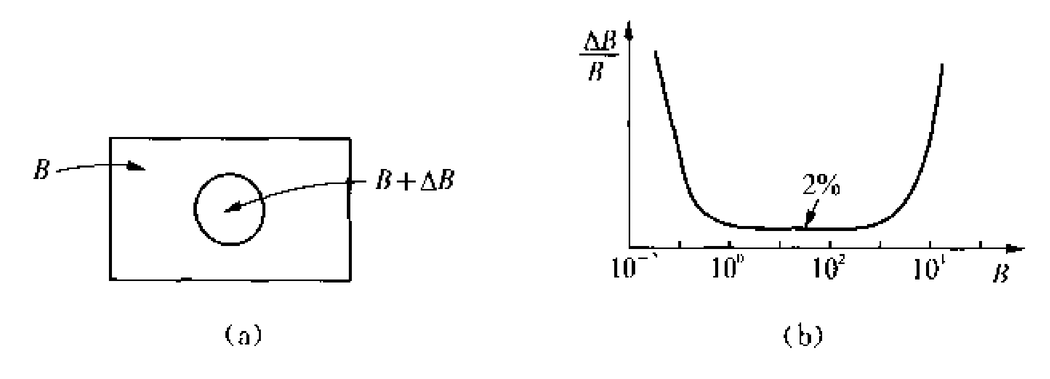


图2.24　具有恒定背景的对比灵敏度

现在根据韦伯定律来建立主观亮度S和亮度B的关系。亮度为B时主观感觉到的亮度为S，可分辨的最小亮度差为ΔB_{min}。ΔB_{min}发生时，主观亮度有变化dS。前面已分析ΔB_{min}越大，敏感性降低，dS就小，因此dS与ΔB_{min}成反比。dS是由B的变化dB引起的，可得

$$dS = k' \frac{1}{\Delta B_{min}} dB = k' \frac{B}{\Delta B_{min}} \frac{dB}{B} = \frac{k'}{\xi} \frac{dB}{B} \tag{2.5-1}$$

对上式积分：

$$S = \frac{k'}{\xi} \ln B + k_0 = \frac{k}{\xi} \ln B + k_0 \tag{2.5-2}$$

式中：k'，k，k_0为常数；在较宽的B的范围内ξ也是常数。上式表明主观亮度S与亮度B的对数成线性关系，这一规律称为韦伯-费赫涅尔定理(Weber-Fechner Law)。当B很高或很低时ξ上升，S随$\lg B$变化的斜率减小，出现了图2.23曲线两端的平坦部分。把式(2.5-2)作成曲线就是图2.23。

但是，在实际条件下观看某一景象时，人眼已适应于某个平均亮度，这时可以分辨的亮度范围就小得多了。为搞清其规律，做了第三个实验。在一个面积较大，亮度为B_0的背景上有亮度为B和$B+\Delta B$的两个目标，如图2.25(a)。以同样方法再一次测出$\frac{\Delta B_{min}}{B}$，而对应每一个参量B_0可得一条曲线，并画于图2.25(b)中。B_0确定时，可分辨的B的范围是在以B_0值为中心的2.2个lg单位内。当B偏离B_0达1.5个lg单位时，主观亮度为“黑”或“白”。与图2.24(b)相比，由于背景B_0的限制，现在对亮度的可分辨范围大大缩小，韦伯比保持常数的范围也大

大缩小了，并可注意到图 2.25(b)各段曲线下面形成的包络就是图 2.24(b)中的曲线。

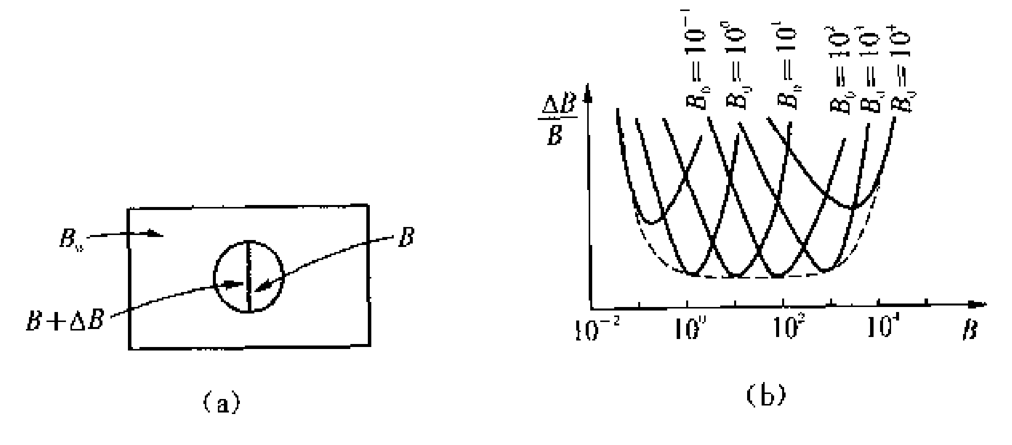


图 2.25 具有可变背景的对比灵敏度

图 2.23 中的虚线曲线段对应着图 2.25(b)中的各条抛物线型曲线。由式(2.5-2)可知：当 $B=B_0$ 时，韦伯比 ξ 最小，斜率 $\frac{k}{\xi}$ 最大，虚线曲线与 $S-\lg B$ 曲线相交。在 B_0 两侧，韦伯比上升，斜率 $\frac{k}{\xi}$ 下降，图 2.23 中虚线曲线的两头较平坦。虚线曲线的左端和右端，分别是主观亮度"全黑"和"全白"。从图中可以看出对于相同的 B，在不同背景亮度 B_0 时，由不同虚线曲线可得出"全白"和"全黑"两种截然不同的主观感觉。

由于人眼适应了某一平均亮度后"黑"和"白"感觉对应的亮度范围缩小，因此在以人眼观察为目的重现数字图像时，重现图像的亮度不必等于原来摄取实际景象时的亮度，只要保持二者的对比度 C 及亮度差别级数（又称灰度层次）相同，就能给人以真实的感觉。对比度 C 定义为

$$C=\frac{B_{max}}{B_{min}} \tag{2.5-3}$$

式中：B_{max} 和 B_{min} 是重现图像（或原景物）的最大和最小亮度。

此外还有所谓相对对比度

$$C_r=\frac{B-B_0}{B_0}\times 100\% \tag{2.5-4}$$

式中：B_0 为背景亮度；B 为物体亮度。

在同样照明条件下，各种物体的亮度决定于物体对入射光的反射能力（即反射系数）。白石膏具有最大反射系数 100%，黑绒有最小反射系数 1%，因而大部分景物的对比度不超过 100。就这个意义上讲，一般图像取样后按 256 级（8 位）灰度层

量化是足够的。

然而如果由人眼来检测某些特定的目标，而目标灰度不易被人眼分辨或在人眼当时的适应性范围之外时，就必须采用图像增强中的一些方法（如伪彩色）来突出该目标，避免因人眼亮度感觉的局限性而漏判。

2.5.4 人眼的分辨率

人眼分辨景物细节的能力称为人眼的分辨率。它包括空间分辨率、亮度分辨率、时间分辨率和彩色分辨率等。

空间分辨率定义为：人眼对被观察物体上能分辨的相邻最近两点的视角 θ 的倒数。其几何关系如图 2.26 所示，其中 L 表示人眼与景物之间的距离；d 表示能分辨的相邻最近两点之间的距离；θ 表示视角，以分为单位。可以得到：

$$\frac{d}{2\pi L} = \frac{\theta(\text{分})}{360 \times 60} \tag{2.5-5}$$

可得

$$\theta = 3\,438\,\frac{d}{L}(\text{分}) \tag{2.5-6}$$

图 2.26 人眼空间分辨率

通常，正常视力在中等亮度和中等相对对比度之下，在观察静止图像时 $\theta \approx 1$ 分。

人眼空间分辨率与下述因素有关：景物成像在黄斑上时，空间分辨率最高；照度太暗，仅杆状细胞起作用，分辨率下降；照度太亮，空间分辨率不会再增加，甚至因“炫目”而降低；相对对比度 C_r 小，即 B 与 B_0 接近，空间分辨率下降。人眼对很细小的“条纹”很难看清楚，就是空间分辨率问题。对于空间分辨率高的部分，要用放大镜才能使人眼看清楚。

亮度分辨率是图像在时间、空间上变化都很缓慢的情况下得到的人眼对亮度变化的分辨能力。人眼对运动很快的目标看不清楚，属于时间分辨率问题。如果时间频率高，则空间分辨率和亮度分辨率都会下降。

空间分辨率、亮度分辨率和时间分辨率是相互联系的，只有三个方面都合适，人眼才能看得清楚。空间分辨率、亮度分辨率和时间分辨率可以进行互换和折衷。

例如,亮度差别小的,如果时间频率很低,则人眼会看清楚,如果时间频率高,则无法看清;亮度差别很大的,即使有轻微晃动,仍能看清楚。当时间频率较高时,人眼对空间对比度的敏感性降低,即对快速运动的物体的细节分辨率降低。当空间频率较高时,人眼对闪烁的敏感度降低。

人眼对彩色空间细节的分辨率低于对黑白画面的分辨率。例如,如果把刚好能分辨的黑白条纹换成亮度相同、颜色不同的彩色条纹时,就不能分辨出条纹来。如果换成红绿相间的条纹,则会由于人眼的空间混色效应而表现出一片黄颜色。实验还表明,人眼对不同的颜色构成的彩色细节的分辨率也不同。表2.3列出了把黑白细节的分辨率定为100%时,人眼对各种彩色细节分辨能力的情况。

表2.3 人眼对彩色细节的分辨能力比较

细节的颜色	黑 白	黑 绿	黑 红	黑 蓝	绿 红	红 蓝	绿 蓝
分辨能力/%	100	94	90	26	40	23	19

另外,人眼对色度的变化比对灰度的变化敏感,因此可用伪彩色技术对图像进行增强,具体将在第四章"图像增强"中介绍。

2.5.5 人眼的适应性

在2.5.2小节中已提到人眼是通过自身的适应性调节,来感觉出宽达10^8倍的亮度范围的。下面再具体介绍一下这种适应性的几个方面:

1. 暗适应

从明亮处走入暗处,视觉要经过几分钟才能适应,约45分钟才能稳定。人眼适应暗环境的功能称为暗适应。这时瞳孔直径可由2 mm扩大到8 mm,进入眼睛的光通量增加到原来的16倍,另一方面由锥状细胞起作用转换为由杆状细胞起作用,后者视敏度为前者的10^4倍。

2. 亮适应

环境由暗变亮时,锥状细胞在几秒钟内就恢复了作用,很快分辨出物体的明暗和颜色。这一适应过程约在3分钟内达到稳定。

3. 局部适应

视网膜上某点或某个局部小区域受强光刺激时,这部分的视敏度就比其他部分的要低。然后再来看均匀亮度的背景时,由于这部分视细胞的视敏度还来不及恢复,就会感到该背景上相应处呈现黑色。

1. 由适应性引起的对比效应

(1) 亮度的同时对比效应。人眼对某个区域感觉到的亮度不是简单地取决于该区域的强度，在相同亮度的刺激下，背景亮度不同时，人眼所感觉到的明暗程度也不同。如图 2.27 所示，三个位于中心的正方形具有完全相同的亮度，而背景具有三个不同的亮度。人眼看上去时，会感觉到暗背景中的正方形看起来要亮些，而亮背景中的正方形看起来要暗些。这是由于当目标被白背景包围时，受目标刺激的视细胞受到了周围的在高亮度光刺激下视细胞视敏度下降的影响，它们所产生的亮度感觉有所下降的缘故。所以，视觉的主观亮度取决于视野中心(目标)与周围环境之间光照的相对强度。

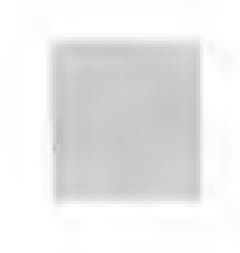
  

图 2.27 同时对比效应

在观察彩色图像时，也有类似的情况，即暗背景中的彩色看起来比亮背景中的彩色明亮一些。除亮度对比效应之外，彩色光还有彩色饱和度对比效应和色调对比效应。

(2) 彩色饱和度对比效应。面积相同、色度与亮度相同的红色区域分别被亮度相同的灰色和红色背景包围，相比之下会感觉红色背景包围的红色区域饱和度较低。

(3) 彩色色调对比效应。面积相同、色度与亮度相同的橘红色区域分别被黄色和红色背景包围，相比之下会感到黄色背景包围的橘红色偏红，而红色背景包围的橘红色偏黄。

(4) 面积对比效应。色度、亮度相同而面积不同的两个彩色区域，面积大的一块会给人以亮度和饱和度都强一些的感觉。

2.5.6 马赫带效应

人眼对于图像上不同空间频率的成分具有不同的灵敏度。实验表明，人眼对于中频的响应较高，对高、低频率的响应较低。由于这一特性，在观察如图 2.28(a)的灰阶条带时，就会显示出马赫带效应。图中是一些不同灰度级的条带，带内亮度

均匀，相邻两条带的亮度之间相差一个固定值。但是看上去每条带的左侧要比右侧亮一些，这就是马赫带效应。由于马赫带效应，相邻条带的边界或图像中亮度的突变处会更加明显。

图2.28(b)中的实线是实际亮度的变化曲线，显示亮度呈现阶梯状的变化。图2.28(b)中的虚线是主观亮度的变化曲线，在亮度阶跃处发生了过冲。在观察图像时，马赫带效应具有增强目标轮廓的作用，由人眼判读图像中的特定目标时是一个有利的因素。

马赫带效应和同时对比效应证明了亮度不是简单的强度函数。

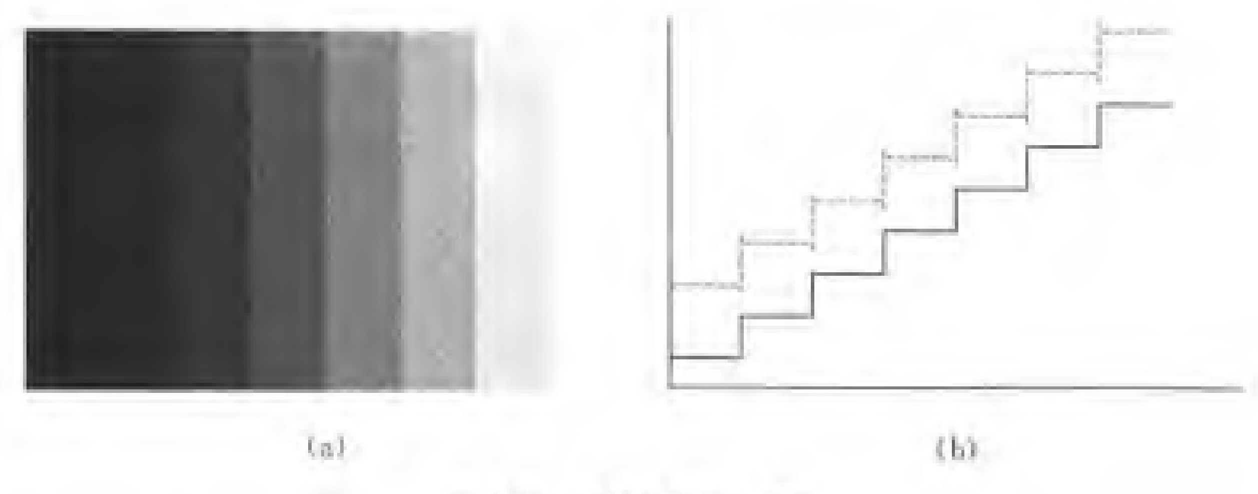


图2.28 马赫带效应示意图

2.5.7 人眼的视觉惰性和闪烁感觉

当有光脉冲刺激人眼时，视觉的建立和消失都需要一定的过程，即具有一定的惰性。光源消失以后，景物影响会在视觉中保留一段时间，称为视觉暂留或视觉惰性现象。视觉暂留时间在0.05～0.2秒。实验表明，若景物以间歇性光亮重复呈现，只要重复频率大于20 Hz，视觉上始终保留有景物存在的印象。

如果周期性重复的脉冲光源作用到视网膜上，当脉冲光的重复频率不够高时，人眼会对之产生一明一暗交替变化的闪烁感觉。这是因为光源在有光和无光间变化时，人眼在亮度感觉上能辨识出它们的差异所致。

2.5.8 视觉的空间错觉

视觉的空间错觉(Space Misconception)是人类视觉系统的一个特性。在错觉中，人眼填充上了不存在的信息或错误地感知物体的几何特点。目前对于人眼的这一特性尚未完全了解。图2.29中给出一些著名的错觉的例子。这些例子可以分为两类，一类是基于形状和方向的，如图2.29(a)、2.29(b)；另一类是基于长度

和面积的，如图2.29(c)、2.29(d)。人眼在观察图2.29(a)时会产生直线有错位的感觉；在图2.29(b)中可看到一个圆；在图2.29(c)中感觉下面的线段比上面的长，而两条线段实际是等长的；在图2.29(d)中，对相同大小的中心圆，人眼看上去却感觉左图中心圆比右图中心圆要小。

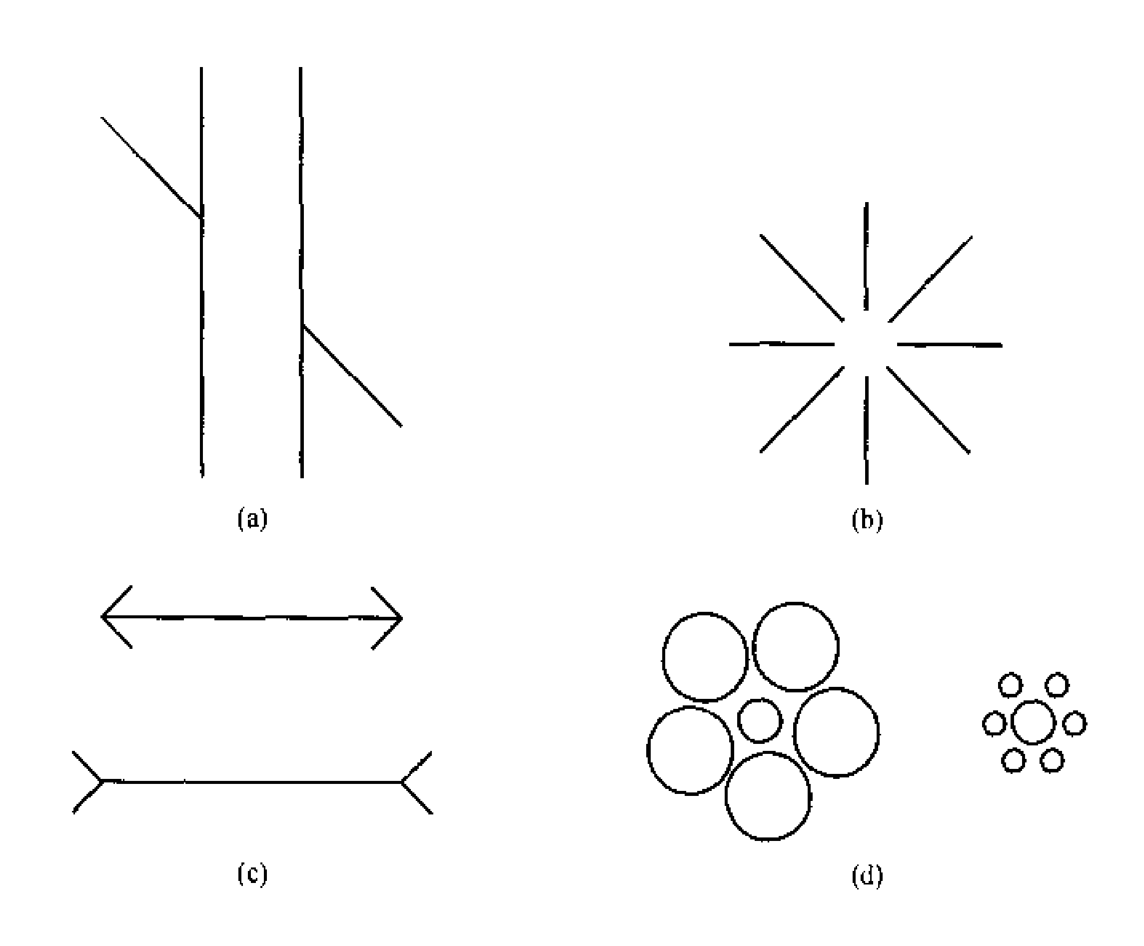


图2.29　一些典型的视觉错觉

2.6　人眼视觉模型与图像质量评价

自20世纪70年代初以来，人们开始关注并探索比较完整的人眼视觉系统(Human Vision System, HVS)理论框架，力求寻找一种基于HVS的客观图像感知质量尺度，架起主客观联系的桥梁。进入20世纪90年代后，数字图像在许多领域获得了广泛的应用，为克服日益增长的巨大数据量与其存储、传输所占用的空间域、时间域和频率域资源有限的矛盾，低比特率有损图像压缩技术需求激增，面新算法所引入的失真更趋多样性、复杂化。与此同时，人们在神经生理学和心理生理学等方面对视觉系统功能的理解有了显著的进展，一大批HVS模型相继出现。相应地，各种图像感知质量尺度应运而生。其性能明显提高。它们作为图像压缩算

法参数最优化和提高像质的目标函数，在各项具体应用中，正逐步取代传统的图像质量评价方法。

2.6.1 人眼视觉模型

典型的HVS模型框架如图2.30所示，它用一个顺序处理过程来模拟人类视觉系统的一系列生理心理反应[5]。

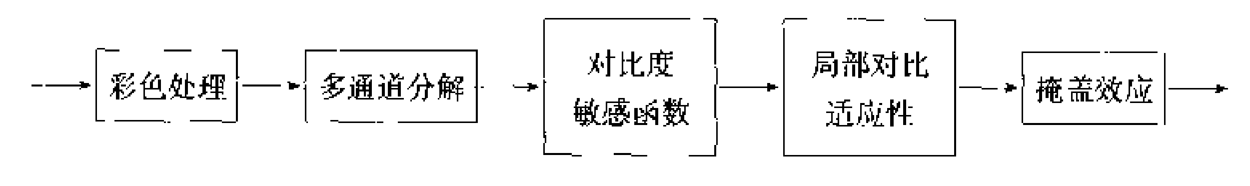


图2.30 一个典型的HVS模型

图2.30中包括了五个处理步骤，分别描述如下：

(1) 彩色处理。HVS模型的第一个步骤是将图像转换到一个适当的感知彩色空间。经过这步转换后，图像用一个亮度分量通道和两个色差分量通道来进行描述。

(2) 多通道分解。对初级视皮层细胞感受器实验测试显示，HVS由相邻的多个并列视觉通道构成。这些通道的空间频率带宽约为一个倍频程，方向上的选择宽度在15°～60°之间。它们具有线性或正交相位、位移不变性、一致的频率响应和多尺度性，对水平(0°)和竖直(90°)方向的刺激最敏感，而向对角方向敏感性逐渐减弱，在45°(或135°)方向处最不敏感。这一结构特征恰好与多分辨滤波器组或小波分解相匹配，使用视皮层变换(Cortex Transform)，即用一组可调的塔式多分辨滤波器建模，可对此进行有效的模拟，而如用小波变换实现，在实际应用中可取得较高的计算效率。

(3) 对比度敏感函数。人眼的光学系统特征以及视网膜等视觉处理过程的非线性与滤波效果，使得对比度阈值(Contrast Threshold)随刺激信号空间频率的不同而变化。对比度阈值的倒数，随空间频率变化的曲线称为对比度敏感函数(Contrast Sensitivity Function, CSF)，CSF经常被看做视觉系统的调制转移函数(Modalation Trausfer Function, MTF)。实际上，MTF对应的只是CSF中的线性滤波成分(光学系统特征)。由于CSF受到诸如方向性、亮度、图像大小及观察距离等多种因素的影响，准确的描述比较复杂，因此在实际应用中，一般认为CSF是带通滤波函数，最大响应区在2～10 c/d(cycle/degree)。当观察距离增大时，低频段的失真将移向最大响应区，因此，CSF有时也采用低通滤波形式。

(4) 局部对比适应性。对人类视觉早期的研究发现，HVS对刺激信号的响

应不是取决于信号的绝对亮度，而是取决于该信号相对于背景亮度（或刺激信号平均亮度）的局部变化，即刺激信号的对比度。人眼主观上刚可察觉的最小亮度差，称为亮度的可见度阈值。对于不同的物理亮度水平，此阈值的响应曲线呈现分段、非线性特点（移变非线性），这就是人眼的亮度适应性，它产生的根源是视细胞非线性地把接收到的光信号转换成电信号。当背景亮度大于 10 cd/m^2 时，亮度适应性可用韦伯定律描述，阈值与背景亮度为对数非线性关系；当背景亮度处于 10 cd/m^2 以下，阈值与背景亮度为平方根（或立方根）关系。由于感知亮度的非线性压缩作用，在高亮度区，视觉感知可以允许较大的图像误差，表现出亮度掩盖效应。

(5) 掩盖效应。掩盖效应，又称视觉掩蔽效应，是指当另一信号出现在一刺激信号的背景中时，刺激信号的可视度降低，即可见度阈值增加。也就是说，刺激信号在超阈值对比度背景下，会被掩盖掉。掩盖效应用阈值提升函数表示（Threshold Elevation, TE），即对比度阈值随背景对比度变化的关系曲线，取决于掩盖物的带宽、相位、方向以及观察者对掩盖物的熟悉程度。根据引发掩盖效应的起因，如强烈的局部对比度、边缘和局部活动性，把它区分为对比度掩盖、边缘掩盖与纹理（噪声）掩盖。由于纹理（噪声）比边缘结构复杂，相对于观察者而言，缺的先验信息更多，因此，纹理（噪声）区的掩盖效应强（TE 曲线斜率增大）。在图像质量评价中，当压缩恢复图像引入超阈值（Super Threshod）失真时，还要注意“互掩盖效应”（Mutual Masking）现象。例如，在原始图像的纹理区域，由于压缩算法引人的失真，可能造成在恢复图像中对应区域的对比度降低，如果使用原始图像作掩盖处理，结果就有偏差。正确的方法是选取两幅图像中掩盖效应小的值作处理。掩盖效应测定的正确与否很大程度上决定了图像感知尺度的应用效果。观察者对图像内容熟悉程度的不确定性以及掩盖效应非局部化的现象给问题的解决带来了困难。

HVS 还有很多种模型，它们从不同的方面反映视觉的某些特性。

图 2.31 是一个对数非线性带通型线性滤波器模型，即对数-线性系统模型，它能在相当宽的亮度范围内预测视觉响应，这种模型较简单，对大多数应用是很合适的。不过在很高空间频率或很强、很弱的亮度下，实际响应与该模型预测数据不符。

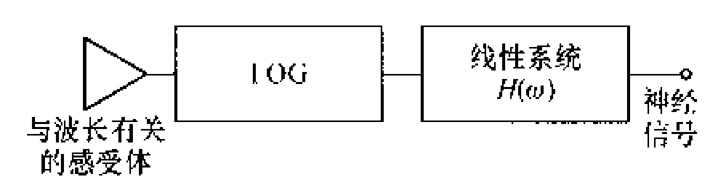


图 2.31 对数-线性系统模型

在图 2.31 中，LOG 是“点”运算，而 $H(u, v)$ 可以用空间域“卷积”或空间频率域“相乘”来表示，也可以用点扩展函数(Point Spread Function, PSF)即 $h(x, y)$ 和/或调制转移函数 MTF 来表示。

对于线性移不变系统，输入二维冲激函数 $\delta(x, y)$，得到输出冲激响应函数 $h(x, y)$，则输入为 $f(x, y)$ 时，输出为 $f(x, y)*h(x, y)$。而 MTF 则是 $h(x, y)$ 的二维傅里叶变换的幅值(模)，即 $|H(u, v)|$。

图 2.32 给出了人眼观察单色图像时的视觉模型，可从四个方面反映出人眼的特性。

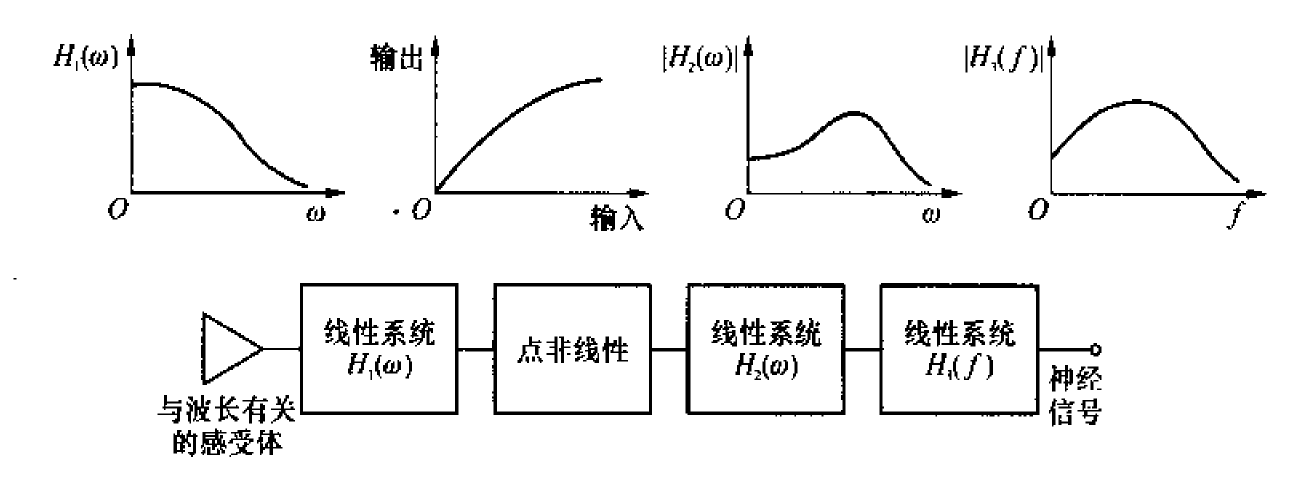


图 2.32 人眼观察单色图像时的视觉模型

(1) 作为一个光学器件，从空间频率域的角度来说，它是一个低通型线性系统。在图 2.32 中以 $H_1(\omega)-\omega$ 曲线表征，其中 $\omega_r^2=u^2+v^2$。这一模型表明人眼的分辨力是有限的。由于瞳孔总有一定的几何尺寸，总有一定的光学像差，视细胞(杆状细胞和锥状细胞)本身有一定的大小，这些因素限制了人眼的分辨力。

(2) 人眼对于亮度的响应具有对数性质，因此它又是一个单调的非线性系统。人眼正是通过这一对数性质来接受宽达 10^8 的亮度范围。图 2.32 中以输入-输出曲线来表征这一对数性质。这种非线性的作用发生在视细胞视觉信号之间的相互作用之前。

(3) 由于神经系统的作用，从空间频率的角度来说，人眼又具有带通型线性系统的特性。在图 2.32 中以 $H_2(\omega)-\omega$ 曲线表征。这种特性又称侧抑制效应，就是说某个视觉信号并不是单纯由一个视细胞感光产生，而是由许多空间上相邻的视细胞的信号加权求和后产生的，即在一定程度上受到周围视细胞的制约，其加权系数如图 2.33 所示。从信号分析理论中可知，对信号进行如图 2.33 那样的加权求和运算，将有带通滤波器的效应。

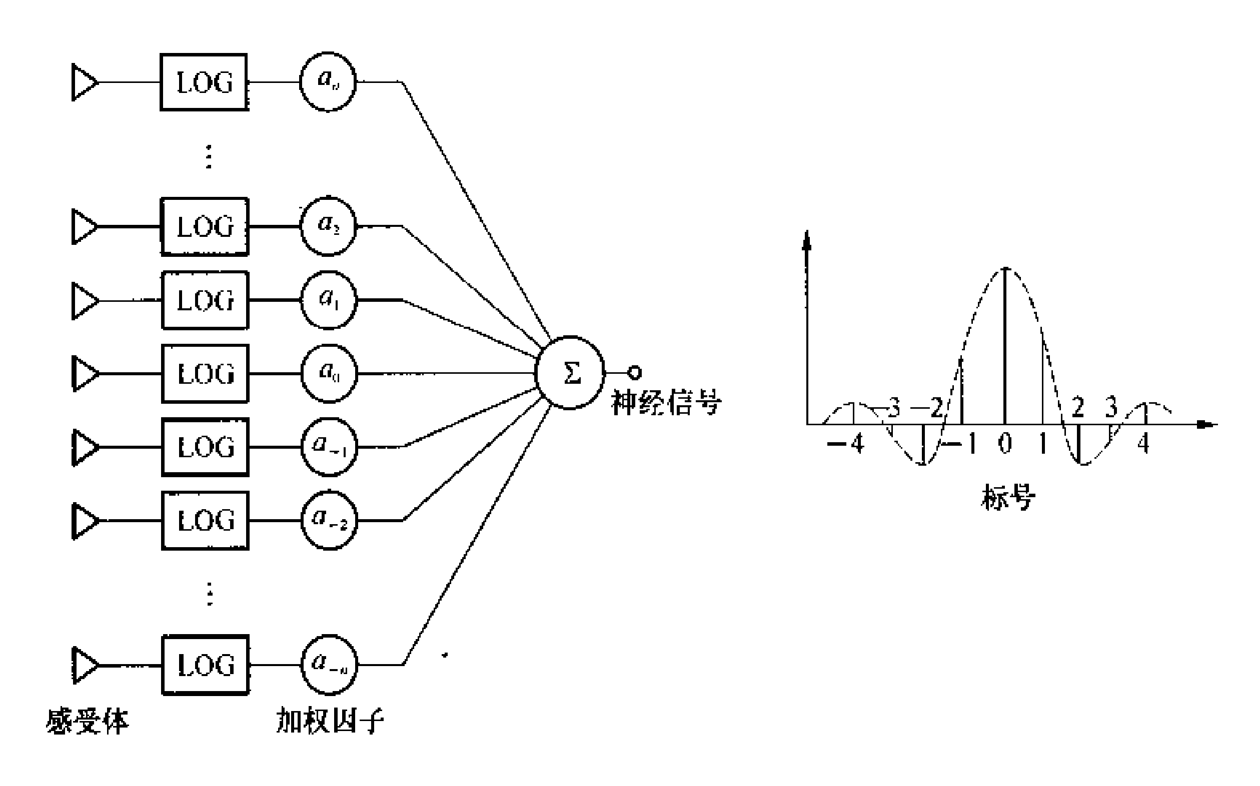


图2.33 侧抑制效应

(4) 从时间上说,人眼对亮度的响应有一定的延时,即当一定亮度的光作用于人眼时,人眼不能在瞬间形成稳定的主观亮度感觉,而有一短暂的过渡过程。同样,光消失后,亮度感觉不是马上消失。这是由神经系统时间响应上的限制引起的,由图2.32中 $H_3(f)-f$ 曲线表征。这是一个模拟时间延迟效应的线性系统。注意,这里是在一个与时间(而不是空间)对应的频率域中描述这种延时特性的。

图2.32中的四条曲线在不同的"域"中描述了系统特性。对于大多数图像处理的实际应用,这一模型是足够详细的,但它没有考虑到人眼的多分辨率特性、方向滤波特性、视觉门限和视觉掩蔽效应。

2.6.2 图像质量评价

在数字图像处理过程中,所有技术的优劣都会影响到图像质量。图像质量的含义主要包括图像的逼真度和可懂度两个方面的内容。为了对图像处理的各个环节进行合理评估,图像质量评价的研究已成为图像信息工程的基础之一。

在传统的图像质量评价方法中,有代表性的方法主要有两种:客观评价和主观评价。

(1) 图像质量的客观评价。对图像质量进行客观评价,实际上是通过与该图像有关的客观参数的大小来反映出图像的质量好坏。在这里介绍几种常用的图像

质量客观评价参数。

峰值信噪比(Peak Signal Noise Ratio，PSNR)。PSNR的定义为：

$$\mathrm{PSNR}=10\log_{10}\frac{255^2}{\frac{1}{MN}\sum_{m=0}^{M-1}\sum_{n=0}^{N-1}[f(m,n)-g(m,n)]^2}\qquad(2.6-1)$$

式中：$f(m,n)$为原图，大小为$M\times N$，灰度取值范围为0～255；$g(m,n)$为处理后的复原图。若$g(m,n)\approx f(m,n)$，则PSNR的值趋于无穷大，此时处理后造成的失真较小，图像质量较好。

由于计算简单，所以用PSNR来判断图像质量的办法用得最多，但PSNR的值较高并不等于图像的主观质量好。例如，当图像的PSNR值很大，但有少量的噪声点出现在图像中的敏感区域，如图中"人脸"上，则虽然PSNR很高，但人眼可看出图像有问题，所以主观质量不好。

均方误差(Mean Square Error，MSE)。MSE的定义为：

$$\mathrm{MSE}=\frac{1}{MN}\sum_{m=0}^{M-1}\sum_{n=0}^{N-1}[f(m,n)-g(m,n)]^2\qquad(2.6-2)$$

当MSE的值越小时，图像的质量就越好，使用时的具体情况与PSNR的情况类似。

关于"与主观评价有较好一致性的客观评价标准"的研究是数字图像处理研究的前沿课题。文献[6]中定义了MPSNR(Masked Peak Signal-to-Noise Ratio)的计算方法，其中"Masked"表示"计入视觉掩蔽效应的"。MPSNR的定义为：

$$\mathrm{MPSNR}=10\lg\frac{255^2}{E^2}\qquad(2.6-3)$$

式中：E^2是考虑了视觉特性后的全图的失真度量，即MPSNR是用分贝表示的计入视觉特性的失真的度量。求MPSNR的过程比较复杂，要先把原图和处理后的图像分解成不同"子带"，分别对应于不同的空间频率和不同的方向。再计算每个子带图像中不同位置处的视觉门限，从而计算出MPSNR。

(2) 图像质量的主观评价。由于在很多应用中，图像最终是给人看的，所以需要以人的主观评价来对图像质量进行判断。在制订一些国际标准(如图像压缩标准)时，最终也是用主观评价的结果来衡量图像质量的好坏。

主观评价结果要受许多因素的影响，这些因素包括观看条件、被测图像序列、评价人和评分结果统计方法等。

主观评价的评分方法包括两类：绝对评价和相对评价。其中，绝对评价是指对每一幅图像或每一个图像序列打分，打分时可参考各种标准图像。相对评价是指

对比一批图像的相对质量好坏，然后进行打分。表 2.4 中给出进行主观质量评价时的评分标准。

表 2.4 图像主观质量评价的评分标准

评 分	绝对评价尺度	相对评价尺度
5	优	一群中最好的
4	良	好于该群中平均水平的
3	中	该群中平均水平
2	差	差于该群中平均水平的
1	劣	该群中最差的

由于图像最终是供人观看的，因此合理评价图像质量的方法应充分考虑到人眼的视觉特性(HVS)。目前，图像质量评价方法的研究从以往物理意义上简单的误差统计方法转为结合人眼视觉特性的误差统计方法。并且，基于 HVS 的测量方法明显优于没有考虑 HVS 的测量方法。基于 HVS 的测量方法主要分为两类：基于视觉感知的测量方法和基于视觉兴趣的测量方法。

基于刚辨差(Just Noticed Difference, JND)的视觉感知测量方法是较早的也是较成功的方法之一。该方法根据人眼分辨亮度差异的能力与背景亮度有关而达到对图像质量定量测量的目的。

人眼在观察和理解图像时往往会对其中某些区域产生兴趣，这些区域被称为“感兴趣区(Region Of Interest, ROI)”。整幅图像的视觉质量往往取决于 ROI 的质量，而不感兴趣区的图像降质常常不易觉察。基于视觉兴趣的图像质量评价方法可通过对 ROI 和不感兴趣区的区别处理，突出人眼对 ROI 的兴趣程度，从而使图像质量的评价结果更符合人眼视觉特性的规律。

习 题

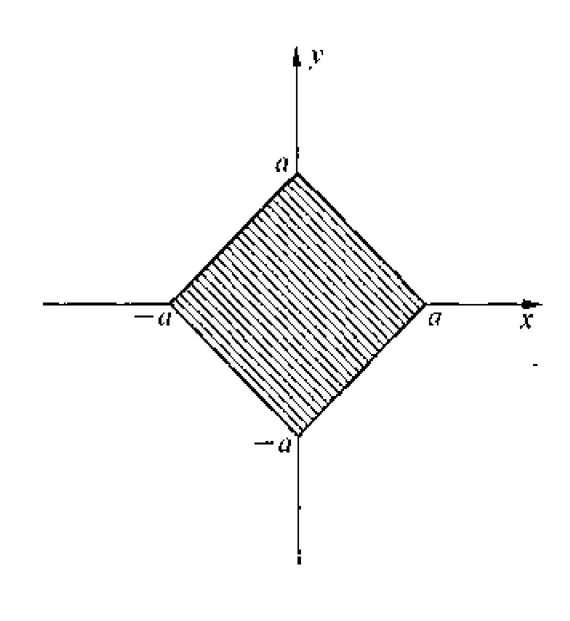
1. 有一线性移不变系统 L，其点扩展函数 $h(x, y)$ 是输入为 $\delta(x)\delta(y)$ 时系统的输出，求下属情况下系统的调制转移函数 $H(u, v)$，

   (1) $h(x, y) = \delta(x - x_0)\delta(y - y_0)$

   (2) $h(x, y) = \begin{cases} E, & |x| \leqslant a \text{ 且 } |y| \leqslant b \\ 0, & \text{其他} \end{cases}$

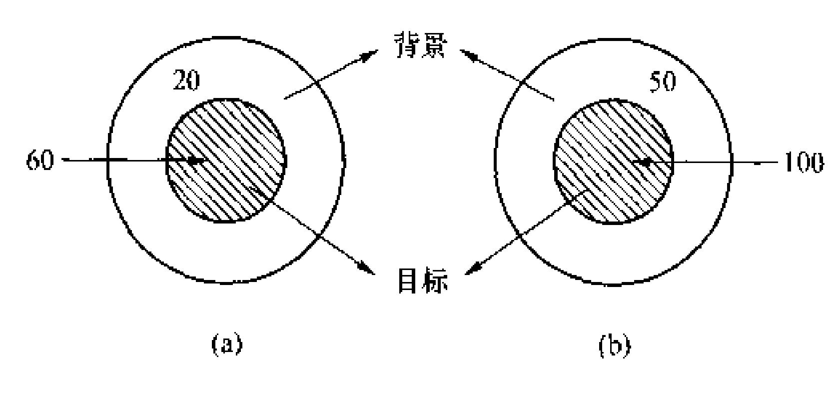
(3) $h(x, y) = \begin{cases} E, & (x, y) \in R \\ 0, & 其他 \end{cases}$

R 如题图 2.1 所示。



题图 2.1

2. 人眼分别观察题图 2.2 所示两幅完全相同的图像，目标与背景的灰度分别如图所标。通过计算判断人眼在观察哪幅图像上的目标时会觉得更亮些（白色灰度值定为 255，黑色灰度值定为 0）。

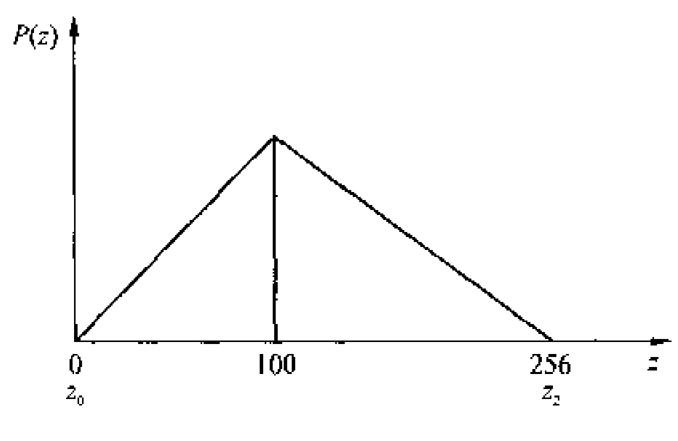


题图 2.2

3. 将模拟图象 $f(x, y)$ 采样成为大小为 256×256 的数字图像 $f(m, n)$，以左上角（$m=n=0$）为原点，各点像素值为 $f(m, n) = 4(m+n) \bmod 256$，若将图象旋转中心设在 $m = n = 128$ 处，逆时针转 $45°$ 后重新采样，得到数字图象 $f'(m, n)$。

求：$f'(127, 127)$ 的值。

4. 若 $P(z)$ 如题图 2.3 所示，试以 Max 量化器的观点将其二值化。即求出 z_1 和



题图 2.3

q_0，q_1（精确到小数点后 1 位）。

5. 数字数据的传输通常用波特率（Baud Rate）来进行度量，当采用二进制时，其定义为每秒钟传输的比特数。通常的传输是以一个开始比特、一个字节（8 比特）的信息和一个停止比特组成的包完成的。基于上述考虑，计算传输下列图像时所需要的时间：

（1）以 56 K 波特的调制解调器传输一幅 256 * 256，256 灰度级的图像；

（2）以 750 K 波特传输一幅 1024 * 1024，16777216 色的真彩色图像。

参考文献

[1] 冈萨雷斯. 数字图像处理(第二版)[M]. 阮秋琦，等译. 北京：电子工业出版社，2003.

[2] Castleman K R. 数字图像处理[M]. 朱志刚，等译. 北京：电子工业出版社，1998.

[3] 罗申菲尔特 A，卡克 A C. 数字图像处理[M]. 余英林，等译. 北京：人民邮电出版社，1982.

[4] Max J. Quantizing for minimum distortion [J]. IRE Trans. Informat. Theory, IT-6, 1960, 7-12.

[5] Marcus J. Nadenu, Stefan Winkler, David Alleysson. Murat Kunt, Human vision models for perceptually optimized image processing — a review.

[6] Al Bovik Ed. Handbook of Image and Video Processing [M]. Academic Press, 2000.

[7] Algazi V R. Useful approximations to optimum quantization [J]. IEEE Trans. Commun., COM-14, 3, June 1966, 297-301.

[8] W. K. 普拉特. 数字图像处理学[M]. 高荣坤，等译. 北京：科学出版社，1984.

[9] Wu Ed H R, Rao K R. Digital video image quality and perceptual coding. CRC Taylor & Francies, 2006.

[10] Hall C F, Hall E L. A nonlinear model for the spatial characteristics of the human visual

system, IEEE Trans. Syst. . Man Cybern. . SMC-7,3, March 1977. 161~170.

[11] Ratliff F, Hartline H K, Miller W H. Spatial and temporal aspects of retinal inhibitory interaction [J]. J. Opt. Soc. Am. . 53,1, January 1963,110-120.

[12] 周新伦,柳健,刘华志. 数字图像处理[M]. 北京:国防工业出版社,1986.

[13] 天津大学电视研究室. 电视原理[M]. 北京:国防工业出版社,1981.

[14] Antoon M van Dijk. et al. . Subjective quality assessment of compressed images [J]. Signal Processing, 1997,58:235~252.

[15] Karunaseka S A. et al. . A distortion measure for blocking artifacts in image based on human visual sensitivity [J]. IEEE Trans. IP, 1995,4(6):713~724.

第3章　图像的二维正交变换

在图像处理中广泛应用着二维正交变换。这是因为利用某些正交变换可以从图像中提取一些特征，如在傅里叶变换后平均值(即“直流”项)正比于图像灰度值的平均值，高频分量则表明了图像中目标边缘的强度及方向。其次是可以在正交变换的基础上，完成图像的变换编码。在变换之后，信号的能量不变，但其分布会有变化，往往会集中到少数一些项上，这时舍弃一些小幅度的变换系数，或对其仅作粗量化时，可以实现数据压缩，有利于存储和传输。在以下各章中二维正交变换被用于图像的增强、复原、编码、描述等方面。

本章讨论傅里叶(Fourier)变换、离散余弦变换、沃尔什(Walsh)变换、离散卡-洛(Karhunen - Loeve)变换以及小波变换。

3.1　正交变换的矩阵表达式

数字图像 $f(m, n)$是具有 M 行 N 列的一个矩阵。若用 $g(m, n; s, t)$表示变换核，则图像从空间域(即 m, n 平面)经某种正交变换转换到变换域(即 s, t 平面)的运算可表达为：

$$F(s, t) = \sum_{m=0}^{M-1}\sum_{n=0}^{N-1} f(m, n) g(m, n; s, t) \tag{3.1-1}$$

式中：m, n 和 s, t 都是整数，它们的取值范围通常是

$$\begin{aligned} &0 \leqslant m, \quad s \leqslant M-1 \\ &0 \leqslant n, \quad t \leqslant N-1 \end{aligned} \tag{3.1-2}$$

变换核 g 有四个变量，不能表示为只有行、列两个方向的矩阵。为了能以矩阵形式表达(3.1-1)的运算，先讨论一些特殊情况。

3.1.1　可分离核

所谓可分离核，是指变换核 g 可以表示为下面的形式：

$$g(m, n; s, t) = u(m, s)v(n, t) \tag{3.1-3}$$

把上式代入(3.1-1),可以把正变换写成:

$$F(s, t) = \sum_{m=0}^{M-1}\left[\sum_{n=0}^{N-1} f(m, n)v(n, t)\right]u(m, s) \tag{3.1-4}$$

这时二维正变换实际上已分解为双重的一维变换,即先以 n 为变量,对图像 $f(m, n)$中的每一行逐行进行一维变换得到一个中间结果 $f'(m, t)$;再对中间结果以 m 为变量,逐列进行第二个一维变换,得到最终的变换结果 $F(s, t)$。

若以矩阵形式表达(3.1-4)式,则有

$$[\boldsymbol{F}] = [\boldsymbol{U}]^{\mathrm{T}}[\boldsymbol{f}][\boldsymbol{V}] \tag{3.1-5}$$

式中:$[f]$是数字图像阵列;$[\boldsymbol{F}]$是经正变换后得到的变换域中的结果,即:

$$[\boldsymbol{F}] = [F(s, t)] = \begin{bmatrix} F(0, 0) & F(0, 1) & \cdots & F(0, N-1) \\ F(1, 0) & F(1, 1) & \cdots & F(1, N-1) \\ \vdots & \vdots & & \vdots \\ F(M-1, 0) & F(M-1, 1) & \cdots & F(M-1, N-1) \end{bmatrix}_{M\times N} \tag{3.1-6a}$$

$$[f] = [f(m, n)] = \begin{bmatrix} f(0, 0) & f(0, 1) & \cdots & f(0, N-1) \\ f(1, 0) & f(1, 1) & \cdots & f(1, N-1) \\ \vdots & \vdots & & \vdots \\ f(M-1, 0) & f(M-1, 1) & \cdots & f(M-1, N-1) \end{bmatrix}_{M\times N} \tag{3.1-6b}$$

$[\boldsymbol{U}]$和$[\boldsymbol{V}]$是变换核 g 分离后所得到的变换矩阵:

$$[\boldsymbol{U}]^{\mathrm{T}} = \begin{bmatrix} u(0, 0) & u(1, 0) & \cdots & u(M-1, 0) \\ u(0, 1) & u(1, 1) & \cdots & u(M-1, 1) \\ \vdots & \vdots & & \vdots \\ u(0, M-1) & u(1, M-1) & \cdots & u(M-1, M-1) \end{bmatrix}_{M\times M} \tag{3.1-6c}$$

$$[\boldsymbol{V}] = \begin{bmatrix} v(0, 0) & v(0, 1) & \cdots & v(0, N-1) \\ v(1, 0) & v(1, 1) & \cdots & v(1, N-1) \\ \vdots & \vdots & & \vdots \\ v(N-1, 0) & v(N-1, 1) & \cdots & v(N-1, N-1) \end{bmatrix}_{N\times N} \tag{3.1-6d}$$

$[\boldsymbol{U}]$和$[\boldsymbol{V}]$矩阵分别是M阶方阵和N阶方阵，每个元素的取值决定于可分离变换核g的数学表达式。注意(3.1-6c)是$[\boldsymbol{U}]$阵的转置$[\boldsymbol{U}]^{\mathrm{T}}$。

如果$[\boldsymbol{U}]$和$[\boldsymbol{V}]$都有逆矩阵在，则从$[\boldsymbol{F}]$到$[\boldsymbol{f}]$的变换可以表达为

$$[\boldsymbol{f}] = ([\boldsymbol{U}]^{\mathrm{T}})^{-1}[\boldsymbol{F}][\boldsymbol{V}]^{-1} \tag{3.1-7}$$

3.1.2　正交变换矩阵

定义：(1) 若M阶实数矩阵满足$[\boldsymbol{U}]\cdot[\boldsymbol{U}]^{\mathrm{T}}=\boldsymbol{I}_M$，则$[\boldsymbol{U}]$称为正交矩阵。

(2) 若M阶复数矩阵满足$[\boldsymbol{U}]([\boldsymbol{U}]^*)^{\mathrm{T}}=\boldsymbol{I}_M$，则$[\boldsymbol{U}]$称为酉矩阵。

下面讨论一下酉矩阵和正交矩阵的几个性质。

性质1：若$[\boldsymbol{U}]$为正交矩阵，则$[\boldsymbol{U}]^{-1}=[\boldsymbol{U}]^{\mathrm{T}}$；

若$[\boldsymbol{U}]$为酉矩阵，则$[\boldsymbol{U}]^{-1}=([\boldsymbol{U}]^*)^{\mathrm{T}}$。

这一性质直接可由定义推得。

性质2(正交归一)：若$[\boldsymbol{U}]$为酉矩阵(或正交矩阵)，则在矩阵$[\boldsymbol{U}]$中各行(或各列)向量的模为1，任意不同行(或不同列)向量之间正交。

证明：矩阵$[\boldsymbol{U}]$可按行或按列，由向量的形式写成。这儿的向量都采用竖排列的方式。

$$[\boldsymbol{U}] = [\boldsymbol{u}_0\ \boldsymbol{u}_1 \cdots \boldsymbol{u}_i \cdots \boldsymbol{u}_{M-1}] = \begin{bmatrix} \boldsymbol{w}_0^{\mathrm{T}} \\ \boldsymbol{w}_1^{\mathrm{T}} \\ \vdots \\ \boldsymbol{w}_i^{\mathrm{T}} \\ \vdots \\ \boldsymbol{w}_{M-1}^{\mathrm{T}} \end{bmatrix} \tag{3.1-8}$$

式中：$\boldsymbol{u}_i$由$[\boldsymbol{U}]$阵中第i列上的各元素组成，而$\boldsymbol{w}_i^{\mathrm{T}}$则由$[\boldsymbol{U}]$阵中第$i$行上的各元素组成，两者都是$M$维向量，即：

$$\boldsymbol{u}_i = [u(0,\ i),\ u(1,\ i)\cdots u(M-1,\ i)]^{\mathrm{T}} \tag{3.1-9a}$$

$$\boldsymbol{w}_i^{\mathrm{T}} = [u(i,\ 0),\ u(i,\ 1)\cdots u(i,\ M-1)] \tag{3.1-9b}$$

当$[\boldsymbol{U}]$为酉矩阵时，根据定义有：

$$[\boldsymbol{U}][\boldsymbol{U}]^{*\mathrm{T}} = \begin{bmatrix} \boldsymbol{w}_0^{\mathrm{T}} \\ \boldsymbol{w}_1^{\mathrm{T}} \\ \vdots \\ \boldsymbol{w}_{M-1}^{\mathrm{T}} \end{bmatrix} [\boldsymbol{w}_0^*\ \boldsymbol{w}_1^*\ \cdots\ \boldsymbol{w}_{M-1}^*] = \begin{bmatrix} 1 & & 0 \\ & \ddots & \\ 0 & & 1 \end{bmatrix} \tag{3.1-10}$$

由上式可得：

$$w_i^{T} \cdot w_j^{*} = \begin{cases} 1, & i = j \\ 0, & i \neq j \end{cases} \tag{3.1-11}$$

同理有

$$[U]^{*T}[U] = \begin{bmatrix} u_0^{*T} \\ u_1^{*T} \\ \vdots \\ u_{M-1}^{*T} \end{bmatrix} [u_0\ u_1\ \cdots\ u_{M-1}] = \begin{bmatrix} 1 & & 0 \\ & \ddots & \\ 0 & & 1 \end{bmatrix} \tag{3.1-12}$$

可得

$$u_i^{*T} u_j = \begin{cases} 1, & i = j \\ 0, & i \neq j \end{cases} \tag{3.1-13}$$

式(3.1-11)和(3.1-13)表明[U]阵是一个正交归一矩阵。

当[U]阵为正交矩阵时，仅去除上述证明中的诸共轭号 *，即可证得它的正交归一性。

证毕。

性质 3：若[U]为酉矩阵，则$[U]^T$ 和$[U]^{-1}$也是酉矩阵。

证明：

(1) $[U]^T([U]^T)^{*T} = [U]^T[U]^* = [([U]^T[U]^*)^T]^T = [[U]^{*T}[U]]^T$
$= [[U]^{-1}[U]]^T = I_M^T = I_M$

因为$[U]^T([U]^T)^{*T} = I_M$，所以$[U]^T$ 是酉矩阵

(2) $[U]^{-1}([U]^{-1})^{*T} = [U]^{-1}([U]^{*T})^{-1} = [U]^{-1} \cdot ([U]^{-1})^{-1} = [U]^{-1}[U] = I_M$

因为$[U]^{-1}([U]^{-1})^{*T} = I_M$，所以$[U]^{-1}$ 是酉矩阵

证毕。

性质 4：若[U]和[V]是阶次相同的酉矩阵，则[U][V]也是酉矩阵。

证明：$([U][V])([U][V])^{*T} = [U][V][V]^{*T}[U]^{*T} = [U][U]^{*T} = I_M$

故[U][V]是酉矩阵。

性质 5：若[U]是酉矩阵，则其行列式的模 $\|[U]\| = 1$。

证明：

$$1 = \|I\| = \|[U][U]^{*T}\| = \left| |[U]| \cdot |[U]^{*T}| \right| = \left| |[U]| \cdot |[U]^{*}| \right|$$

$$= |(a+bj)(a-bj)| = |a+bj||a-bj| = |a+bj| \cdot |a+bj| = \|[U]\|^2$$

得到 $\|[U]\|^2 = 1$　　即 $\|[U]\| = 1$

证毕。

在上述证明中，由于$[U]$为一个复数方阵，故$|[U]|$也是复数，则令$|[U]|=a+bj$，可推出

$$|[U]^*|=a-bj,$$

式中：a和b是实数，j是虚数单位符号。

性质6：若$[U]$是酉矩阵，$\boldsymbol{a}$是向量，作变换$\boldsymbol{b}=[U]\boldsymbol{a}$，则向量的模有$|\boldsymbol{a}|=|\boldsymbol{b}|$。

证明：

$$|\boldsymbol{b}|^2=\boldsymbol{b}^{*\mathrm{T}}\boldsymbol{b}=([U]\boldsymbol{a})^{*\mathrm{T}}([U]\boldsymbol{a})=\boldsymbol{a}^{*\mathrm{T}}[U]^{*\mathrm{T}}[U]\boldsymbol{a}=\boldsymbol{a}^{*\mathrm{T}}\boldsymbol{a}=|\boldsymbol{a}|^2$$

得到$|\boldsymbol{b}|=|\boldsymbol{a}|$。

在3.1.1小节中已推得的矩阵形式表达的正变换式(3.1-5)和反变换式(3.1-7)中，当$[\boldsymbol{U}]$和$[\boldsymbol{V}]$都是酉矩阵时，则以矩阵形式表达的正变换和反变换为

$$\begin{aligned}[\boldsymbol{F}]&=[\boldsymbol{U}]^{\mathrm{T}}[\boldsymbol{f}][\boldsymbol{V}]\\ [\boldsymbol{f}]&=[\boldsymbol{U}]^*[\boldsymbol{F}][\boldsymbol{V}]^{*\mathrm{T}}\end{aligned} \tag{3.1-14}$$

当$[\boldsymbol{U}]$和$[\boldsymbol{V}]$都是正交变换矩阵时，式(3.1-14)更进一步简化为

$$[\boldsymbol{f}]=[\boldsymbol{U}][\boldsymbol{F}][\boldsymbol{V}]^{\mathrm{T}}$$

3.1.3 图像矩阵的外积分解

$\boldsymbol{a}^{\mathrm{T}}\boldsymbol{b}$称为向量$\boldsymbol{a}$和$\boldsymbol{b}$的内积，它是一个标量；$\boldsymbol{a}\boldsymbol{b}^{\mathrm{T}}$称为向量$\boldsymbol{a}$和$\boldsymbol{b}$的外积，外积的结果是一个矩阵。经过图像矩阵的外积分解后，可以把图像矩阵看成$M\times N$个基本图像加权求和的结果，加权因子是$F(s,t)$。

把(3.1-14)改写一下：

$$[\boldsymbol{f}]=[\boldsymbol{U}]^*[\boldsymbol{F}][\boldsymbol{V}]^{*\mathrm{T}}=[\boldsymbol{u}_0^*\ \boldsymbol{u}_1^*\ \cdots\ \boldsymbol{u}_{M-1}^*][\boldsymbol{F}]\begin{bmatrix}\boldsymbol{v}_0^{*\mathrm{T}}\\ \boldsymbol{v}_1^{*\mathrm{T}}\\ \vdots\\ \boldsymbol{v}_{N-1}^{*\mathrm{T}}\end{bmatrix} \tag{3.1-15}$$

把$[\boldsymbol{F}]$分解成$M\times N$个矩阵求和的形式，使每个分矩阵中只有一个非零元素：

$$[\boldsymbol{F}]=\begin{bmatrix}F(0,0) & F(0,1) & \cdots & F(0,N-1)\\ F(1,0) & F(1,1) & \cdots & F(1,N-1)\\ \vdots & \vdots & & \vdots\\ F(M-1,0) & F(M-1,1) & \cdots & F(M-1,N-1)\end{bmatrix}$$

$$
=\begin{bmatrix} F(0,0) & 0 & \cdots & 0 \\ 0 & 0 & \cdots & 0 \\ \vdots & \vdots & & \vdots \\ 0 & 0 & \cdots & 0 \end{bmatrix}+\begin{bmatrix} 0 & F(0,1) & \cdots & 0 \\ 0 & 0 & \cdots & 0 \\ \vdots & \vdots & & \vdots \\ 0 & 0 & \cdots & 0 \end{bmatrix}
$$

$$
+\cdots+\begin{bmatrix} 0 & 0 & \cdots & 0 \\ 0 & 0 & \cdots & 0 \\ \vdots & \vdots & & \vdots \\ 0 & 0 & \cdots & F(M-1,N-1) \end{bmatrix} \tag{3.1-16}
$$

将式(3.1-16)代入式(3.1-15)中，则包含着[$\boldsymbol{F}$]阵中第s行t列的元素$F(s,t)$的分矩阵对应着

$$
[\boldsymbol{u}_0^* \cdots \boldsymbol{u}_s^* \cdots \boldsymbol{u}_{M-1}^*]\begin{bmatrix} 0 & \cdots & \cdots & 0 \\ \vdots & F(s,t) & & \vdots \\ \vdots & & & \vdots \\ 0 & \cdots & \cdots & 0 \end{bmatrix}\begin{bmatrix} \boldsymbol{v}_0^{*\mathrm{T}} \\ \vdots \\ \boldsymbol{v}_t^{*\mathrm{T}} \\ \vdots \\ \boldsymbol{v}_{N-1}^{*\mathrm{T}} \end{bmatrix}=\boldsymbol{u}_s^* F(s,t)\boldsymbol{v}_t^{*\mathrm{T}}
$$

$$
=F(s,t)\boldsymbol{u}_s^* \boldsymbol{v}_t^{*\mathrm{T}} \tag{3.1-17}
$$

因此，式(3.1-15)可进一步写成：

$$
[\boldsymbol{f}]=\sum_{s=0}^{M-1}\sum_{t=0}^{N-1}F(s,t)\boldsymbol{u}_s^* \boldsymbol{v}_t^{*\mathrm{T}} \tag{3.1-18}
$$

$\boldsymbol{u}_s^* \boldsymbol{v}_t^{*\mathrm{T}}$是两个向量的外积，产生一个矩阵，又称基本图像。若图像矩阵有M行N列，则共有$M\times N$个基本图像，每个基本图像也是M行N列。在某些变换中(如傅里叶变换和余弦变换)，当M和N确定后，基本图像也就确定了，与[$\boldsymbol{f}$]无关；但在另外一些变换(如卡-洛变换及奇异值分解)中，基本图像与[$\boldsymbol{f}$]有关。$F(s,t)$是基本图像$\boldsymbol{u}_s^* \boldsymbol{v}_t^{*\mathrm{T}}$的加权因子，图像阵列[$\boldsymbol{f}$]是基本图像加权求和的结果。

3.1.4 堆叠

变换核$g(m,n;s,t)$不可分离时，无法产生(3.1-6c)和(3.1-6d)那样形式的变换矩阵，得不到(3.1-5)那样的矩阵形式的变换式。如果把图像阵列转换成向量的形式，数据结构就从二维变成了一维，那么原来四个变量的变换核就可以用矩阵(二维)的形式来表达。这样，即使变换核不可分离，变换过程也可用矩阵-向量形式来表达。

对图像阵列[$\boldsymbol{f}$]中的诸元素按行的顺序，逐行首尾相接，把全部$M\times N$个元素

堆成一个列向量 $\boldsymbol{f}$ 如下：

$$\boldsymbol{f}=[\boldsymbol{f}_0^{\mathrm{T}}\ \boldsymbol{f}_1^{\mathrm{T}}\ \cdots\ \boldsymbol{f}_{M-1}^{\mathrm{T}}]^{\mathrm{T}}=[f_0,\ f_1,\ \cdots,\ f_{MN-1}]^{\mathrm{T}} \qquad (3.1-19)$$

式中：$\boldsymbol{f}_i$ 是第 i 行的 N 个元素按原顺序排成的列向量，即：

$$\boldsymbol{f}_i=[f(i,\ 0)\ f(i,\ 1)\ \cdots\ f(i,\ N-1)]^{\mathrm{T}} \qquad (3.1-20)$$

这一处理过程称为堆叠，参见图 3.1。

$\boldsymbol{f}$ 中共有 M 组数据，每组 N 个元素，其中第 k 个元素：

$$\begin{cases} f_k=f(m,\ n), \\ k=mN+n, \end{cases} \quad 0\leqslant k\leqslant MN-1 \qquad (3.1-21)$$

对变换域矩阵[$\boldsymbol{F}$]的处理也类似，即

$$\boldsymbol{F}=[\boldsymbol{F}_0^{\mathrm{T}}\ \boldsymbol{F}_1^{\mathrm{T}}\ \cdots\ \boldsymbol{F}_{M-1}^{\mathrm{T}}]^{\mathrm{T}}=[F_0,\ F_1,\ \cdots,\ F_{MN-1}]^{\mathrm{T}} \qquad (3.1-22)$$

其中第 l 个元素：

$$\begin{cases} F_l=F(s,\ t), \\ l=sN+t, \end{cases} \quad 0\leqslant l\leqslant MN-1 \qquad (3.1-23)$$

这儿用 $\boldsymbol{F}$ 表示一个向量。

正变换可写成

$$\boldsymbol{F}=[\boldsymbol{G}]^{\mathrm{T}}\boldsymbol{f} \qquad (3.1-24)$$

图 3.1　堆叠

变换矩阵[$\boldsymbol{G}$]为

$$[\boldsymbol{G}]=[g_{k,l}]=\begin{bmatrix} g_{0,0} & g_{0,1} & \cdots & g_{0,MN-1} \\ g_{1,0} & g_{1,1} & \cdots & g_{1,MN-1} \\ \vdots & \vdots & & \vdots \\ g_{MN-1,0} & g_{MN-1,1} & \cdots & g_{MN-1,MN-1} \end{bmatrix}_{MN\times MN} \qquad (3.1-25)$$

式中：

$$\begin{cases} k=mN+n, \\ l=sN+t, \end{cases} \quad 0\leqslant k,\ l\leqslant MN-1 \qquad (3.1-26)$$

式(3.1-25)建立了[$\boldsymbol{G}$]的元素与原变换核 $g(m,\ n;\ s,\ t)$之间的关系。

3.1.5 矩阵的直积(Kronecker 积)

定义:设 $[\boldsymbol{A}]=[a_{kl}]$ 是 K 行 L 列矩阵,$[\boldsymbol{B}]=[b_{mn}]$ 是 M 行 N 列矩阵,有直积 $[\boldsymbol{C}]=[\boldsymbol{A}]\otimes[\boldsymbol{B}]$,以分块矩阵的形式,$[\boldsymbol{C}]$可以表示为

$$[\boldsymbol{C}]=[\boldsymbol{A}]\otimes[\boldsymbol{B}]=\begin{bmatrix} a_{00}[B] & a_{01}[B] & \cdots & a_{0,L-1}[B] \\ a_{10}[B] & a_{11}[B] & \cdots & a_{1,L-1}[B] \\ \vdots & \vdots & & \vdots \\ a_{K-1,0}[B] & a_{K-1,1}[B] & \cdots & a_{K-1,L-1}[B] \end{bmatrix}_{KM\times LN} \tag{3.1-27}$$

$[\boldsymbol{C}]$的元素

$$C_{km,ln}=a_{kl}b_{mn} \tag{3.1-28}$$

式中:

$$\begin{cases} km=kM+m, & 0\leqslant km\leqslant KM-1 \\ ln=lN+n, & 0\leqslant \ln\leqslant LN-1 \end{cases}$$

而 $0\leqslant k\leqslant K-1$, $0\leqslant l\leqslant L-1$, $0\leqslant m\leqslant M-1$, $0\leqslant n\leqslant N-1$。

矩阵的直积具有下面的性质,并假定已满足了某些运算可以进行的必要前提:

性质 1: $([\boldsymbol{A}]\otimes[\boldsymbol{B}])([\boldsymbol{C}]\otimes[\boldsymbol{D}])=([\boldsymbol{A}][\boldsymbol{C}])\otimes([\boldsymbol{B}][\boldsymbol{D}])$ (3.1-29)

性质 2: $T_r([\boldsymbol{A}]\otimes[\boldsymbol{B}])=(T_r[\boldsymbol{A}])(\boldsymbol{T}_r[\boldsymbol{B}])$ (3.1-30)

性质 3: $([\boldsymbol{A}]+[\boldsymbol{B}])\otimes[\boldsymbol{C}]=([\boldsymbol{A}]\otimes[\boldsymbol{C}])+([\boldsymbol{B}]\otimes[\boldsymbol{C}])$ (3.1-31)

性质 4: $([\boldsymbol{A}]\otimes[\boldsymbol{B}])^{-1}=[\boldsymbol{A}]^{-1}\otimes[\boldsymbol{B}]^{-1}$ (3.1-32)

性质 5: $([\boldsymbol{A}]\otimes[\boldsymbol{B}])^{\mathrm{T}}=[\boldsymbol{A}]^{\mathrm{T}}\otimes[\boldsymbol{B}]^{\mathrm{T}}$ (3.1-33)

利用矩阵直积的概念,可以用堆叠的形式表达可分离变换核下的变换表达式。根据式(3.1-5),以矩阵下标的形式并按矩阵乘法定义写出计算$[\boldsymbol{F}]$中每个元素的表达式为

$$F_{st}=\sum_{m=0}^{M-1}\sum_{n=0}^{N-1}([\boldsymbol{U}]^{\mathrm{T}})_{sm}f_{mn}[V_{nt}]=\sum_{m=0}^{M-1}\sum_{n=0}^{N-1}[U_{ms}][V_{nt}]f_{mn}$$

由式(3.1-28)可写成

$$\begin{aligned} F_{st} &= \sum_{m=0}^{M-1}\sum_{n=0}^{N-1}([\boldsymbol{U}]\otimes[\boldsymbol{V}])_{mn,st}f_{mn}=\sum_{m=0}^{M-1}\sum_{n=0}^{N-1}[([\boldsymbol{U}]\otimes[\boldsymbol{V}])^{\mathrm{T}}]_{st,mn}f_{mn} \\ &= \sum_{m=0}^{M-1}\sum_{n=0}^{N-1}([\boldsymbol{U}]^{\mathrm{T}}\otimes[\boldsymbol{V}]^{\mathrm{T}})_{st,mn}f_{mn} \end{aligned}$$

根据下标转换式(3.1-26),上式可改写为

$$F_l=\sum_{k=0}^{MN-1}([\boldsymbol{U}]^{\mathrm{T}}\otimes[\boldsymbol{V}]^{\mathrm{T}})_{lk}f_k \tag{3.1-34}$$

式(3.1-34)是矩阵向量相乘时元素的表达式,因而对应式(3.1-5)有

$$\boldsymbol{F}=([\boldsymbol{U}]^{\mathrm{T}}\otimes[\boldsymbol{V}]^{\mathrm{T}})\boldsymbol{f} \tag{3.1-35}$$

式中:$\boldsymbol{F}$、$\boldsymbol{f}$ 的含义同式(3.1-22)和(3.1-19)。

3.1.6 二维正交变换下的能量守恒

如果正、反变换核是可分离的,且变换矩阵是酉矩阵或正交矩阵,则所进行的变换是正交变换,可以用矩阵形式表示为

$$[\boldsymbol{F}]=[\boldsymbol{U}]^{\mathrm{T}}[\boldsymbol{f}][\boldsymbol{V}]$$

$$[\boldsymbol{f}]=[\boldsymbol{U}]^{*}[\boldsymbol{F}][\boldsymbol{V}]^{*\mathrm{T}}$$

此时下述等式将成立:

$$\sum_{m=0}^{M-1}\sum_{n=0}^{N-1}|f(m,n)|^2=\sum_{s=0}^{M-1}\sum_{t=0}^{N-1}|F(s,t)|^2 \tag{3.1-36}$$

即变换域中信号能量与原来空间域中的信号能量相等。下面对式(3.1-36)作一证明。

若以 $T_r\{\cdot\}$表示求迹运算,即对某一方阵对角线上的元素作求和运算,有

$$\sum_{s=0}^{M-1}\sum_{t=0}^{N-1}|F(s,t)|^2=T_r\{[\boldsymbol{F}]^{\mathrm{T}}[\boldsymbol{F}]^{*}\}$$

将式(3.1-5)、(3.1-14)代入上式,并考虑到$\boldsymbol{U}$, $\boldsymbol{V}$ 为酉阵:

$$\begin{aligned}\sum_s\sum_t|F(s,t)|^2&=T_r\{([\boldsymbol{U}]^{\mathrm{T}}[\boldsymbol{f}][\boldsymbol{V}])^{\mathrm{T}}([\boldsymbol{U}]^{\mathrm{T}}[\boldsymbol{f}][\boldsymbol{V}])^{*}\}\\&=T_r\{[\boldsymbol{V}]^{\mathrm{T}}[\boldsymbol{f}]^{\mathrm{T}}[\boldsymbol{f}]^{*}[\boldsymbol{V}]^{*}\}\end{aligned}$$

由矩阵乘法规则,同时进行求迹运算,上式可写成:

$$\sum_s\sum_t|F(s,t)|^2=\sum_k\sum_l\sum_p\{([\boldsymbol{V}]^{\mathrm{T}})_{kl}([\boldsymbol{f}]^{\mathrm{T}}[\boldsymbol{f}]^{*})_{lp}([\boldsymbol{V}]^{*})_{pk}\}$$

为简洁起见,上式中用双下标表示矩阵中的某个元素。这里每个元素是标量,可以交换次序,并对其中的$([\boldsymbol{V}]^{\mathrm{T}})_{kl}$和$([\boldsymbol{V}]^{*})_{pk}$作转置处理:

$$\sum_s\sum_t|F(s,t)|^2=\sum_k\sum_l\sum_p\{([\boldsymbol{f}]^{\mathrm{T}}[\boldsymbol{f}]^{*})_{lp}([\boldsymbol{V}])_{lk}([\boldsymbol{V}]^{*\mathrm{T}})_{kp}\}$$

$$
\begin{aligned}
&= \sum_l \sum_p \{([\boldsymbol{f}]^{\mathrm{T}}[\boldsymbol{f}]^*)_{lp} \sum_k [([\boldsymbol{V}])_{lk}([\boldsymbol{V}]^{*\mathrm{T}})_{kp}]\} \\
&= \sum_l \sum_p \{([\boldsymbol{f}]^{\mathrm{T}}[\boldsymbol{f}]^*)_{lp}([\boldsymbol{V}][\boldsymbol{V}]^{*\mathrm{T}})_{lp}\} \\
&= \sum_l \sum_p \{([\boldsymbol{f}]^{\mathrm{T}}[\boldsymbol{f}]^*)_{lp}(\boldsymbol{I})_{lp}\} \\
&= \sum_l \{([\boldsymbol{f}]^{\mathrm{T}}[\boldsymbol{f}]^*)_{ll}\} \\
&= T_r\{[\boldsymbol{f}]^{\mathrm{T}}[\boldsymbol{f}]^*\}
\end{aligned}
$$

最后得到

$$
\sum_s \sum_t |F(s, t)|^2 = \sum_m \sum_n |f(m, n)|^2
$$

证毕。

虽然式(3.1-36)反映了变换前后信号能量不变，但是在$[\boldsymbol{f}]$和$[\boldsymbol{F}]$中能量的分布情况却不同。如果在$[\boldsymbol{F}]$中，信号能量集中在少数几项上(一般情况下都是这样)，则可用较少的数据量代替原图，达到压缩数据量的目的。当然这样做会引入一定的误差，以后我们还将详细讨论这个问题。

3.2 傅里叶变换

傅里叶变换的研究与应用已有很长的历史。六十年代中期在计算机上实现了离散傅里叶变换的快速算法(即快速傅里叶变换算法FFT)，才使其研究与应用取得了突破性的进展。

某些数字图像处理的算法(如下一章的锐化和平滑)、经傅里叶变换后在频率域上进行分析，可以得到更深入、更具体的结论，而由单纯空间域上的分析往往不容易得到这些结论。在特征提取与数据压缩方面，傅里叶变换的应用也有其独到之处。本节先讨论连续图像$f(x, y)$的变换，再讨论数字图像$f(m, n)$的变换及其性质。

3.2.1 连续图像 $f(x, y)$ 的傅里叶变换

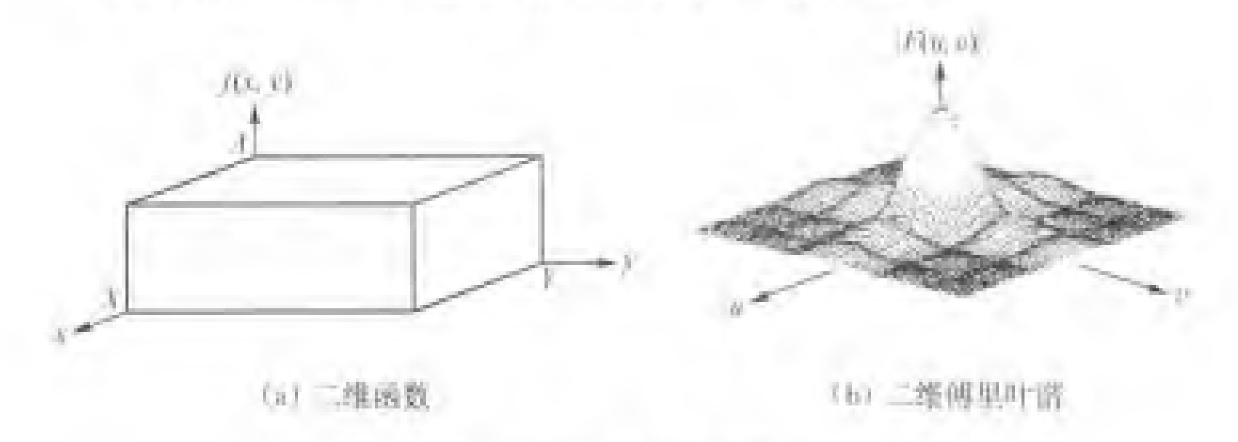
如第2章中所述，连续图像$f(x, y)$可看做$x-y$平面上的一个函数。如果$f(x, y)$是连续的和可积的，则它的傅里叶变换为：

$$
F(u, v) = \iint_{-\infty}^{\infty} f(x, y)\exp[-j2\pi(ux + vy)]\mathrm{d}x\mathrm{d}y \qquad (3.2-1)
$$

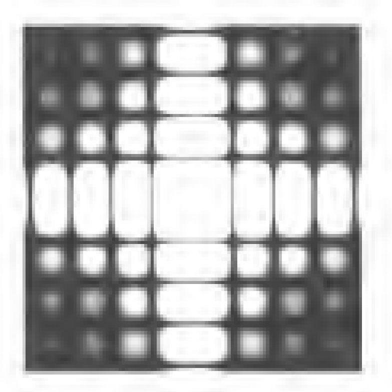
对应的反变换表达式为：

$$f(x, y) = \iint_{-\infty}^{\infty} F(u, v)\exp[j2\pi(ux + vy)]\mathrm{d}u\mathrm{d}v \qquad (3.2-2)$$

图3.2中示出了一个二维函数及其二维傅里叶变换的谱。



（a）二维函数　　　　（b）二维傅里叶谱



（c）作为强度函数显示的谱

图3.2　二维傅里叶谱示例

由欧拉公式，式(3.2-2)可以改写为：

$$f(x, y) = \iint_{-\infty}^{\infty} F(u, v)[\cos 2\pi(ux + vy) + \mathrm{j}\sin 2\pi(ux + vy)]\mathrm{d}u\mathrm{d}v \qquad (3.2-3)$$

上式表明图像 $f(x, y)$ 可以看成由无数正弦和余弦函数加权求和得到，加权因子为 $F(u, v)$。

取其中的 $\cos 2\pi(ux + vy)$ 来分析，见图3.3。u 和 v 取固定值，令

$$ux + vy = n \qquad (3.2-4)$$

当 $n = 0, \pm 1, \pm 2, \cdots$ 时，$\cos 2\pi(ux + vy) = 1$（最大值）。所有满足式(3.2-4)的点 (x, y) 构成了图(a)中的实线，它们是无数的斜平行线。根据式(3.2-4)，它们与 x、y 轴的截距分别为：

$$x = \frac{n}{u}, \quad y = \frac{n}{v} \tag{3.2-5}$$

由式(3.2-5)可求出相邻平行线之间的距离(即空间周期)

$$T = \frac{1}{\sqrt{u^2 + v^2}} \tag{3.2-6}$$

因此余弦函数 $\cos 2\pi(ux + vy)$ 的空间频率

$$\frac{1}{T} = \sqrt{u^2 + v^2} \tag{3.2-7}$$

通常 u 和 v 分别表示 $\cos 2\pi(ux + vy)$ 沿 x 和 y 向的空间频率的分量。

同时可以求出平行线与 x 轴的夹角

$$\alpha = 90^\circ + \arctan \frac{v}{u} \tag{3.2-8}$$

由式(3.2-6)和(3.2-8)可知，当 u、v 取的值不同时，平行线之间的距离 T 及其与 x 轴的夹角 α 都会有变化。对 $\sin 2\pi(ux + vy)$ 的分析也类似。

这一分析表明，图像 $f(x, y)$ 可以由无数的二维正弦和余弦基图像线性组合而成，而加权因子即为 $F(u, v)$。$F(u, v)$ 称为 $f(x, y)$ 的空间频谱。

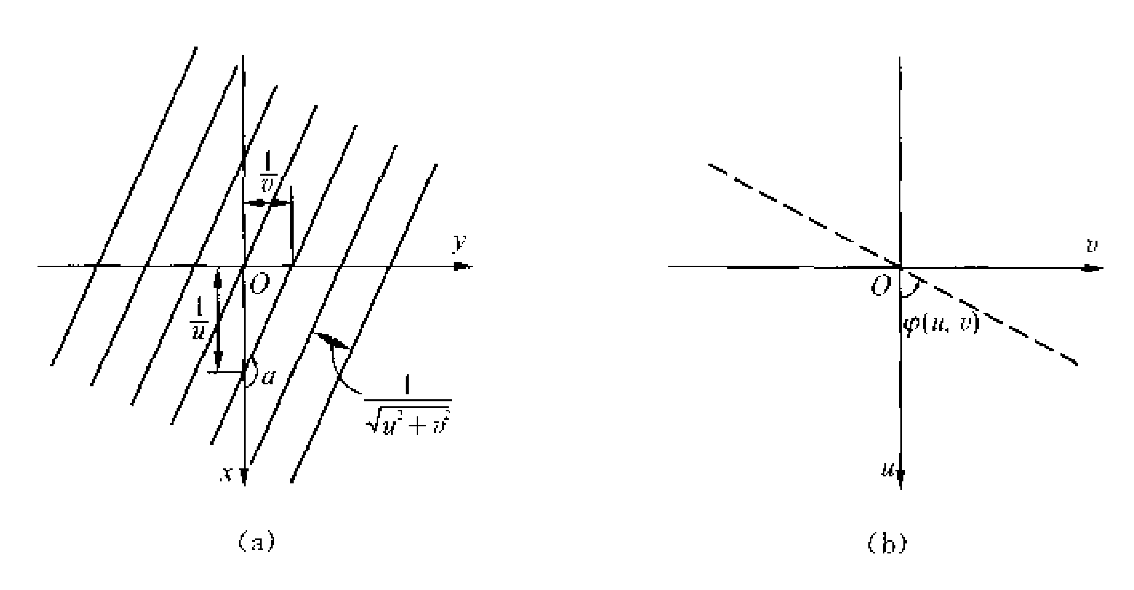


图 3.3 余弦图像及其空间频谱

在图像信号中，空间频率是指在单位长度内亮度(或灰度值)作周期性变化的次数，其单位为线对/毫米。空间频率向量可由 $ui + vj$ 表示，i 和 j 分别是沿 u 轴和 v 轴的单位向量。该向量的幅值(即空间频率)由(3.2-7)式确定，向量与 u 轴的夹角

$$\varphi = \arctan \frac{v}{u} = \alpha - 90^\circ \tag{3.2-9}$$

如图 3.3(b)所示。可见当各坐标轴方向如图那样确定时，空间频率向量与相应的 xy 平面上的平行线垂直。

当图像中目标的某个边缘是一段与 x 轴成 α 角的直线时，这个边缘就是由许多方向与 x 轴成 α 角而空间周期不同的二维余弦(正弦)图像线性组合而成，它们的空间频率分量之比 $\frac{v}{u}=$ 常数。这样的空间频谱由一系列点构成，这些点排在一条直线上，该直线与 u 轴的夹角为 φ，如图 3.3(b)中的虚线所示。

3.2.2 数字图像 $f(m, n)$ 的离散傅里叶变换

1. 二维离散傅里叶变换 DFT 表达式的结构

数字图像 $f(m, n)$ 是以一个 M 行 N 列的矩阵 $[f(m, n)]_{M\times N}$ 的形式出现在计算机中的。图像矩阵的二维离散傅里叶变换可按级数形式定义为：

$$F(s, t)=\frac{1}{\sqrt{MN}}\sum_{m=0}^{M-1}\sum_{n=0}^{N-1}f(m, n)\exp\left[-\mathrm{j}2\pi\left(\frac{sm}{M}+\frac{tn}{N}\right)\right] \quad (3.2-10)$$

式中：$s=0, 1, \cdots, M-1$；$t=0, 1, \cdots, N-1$。

离散傅里叶反变换式可定义为：

$$f(m, n)=\frac{1}{\sqrt{MN}}\sum_{s=0}^{M-1}\sum_{t=0}^{N-1}F(s, t)\exp\left[\mathrm{j}2\pi\left(\frac{sm}{M}+\frac{tn}{N}\right)\right] \quad (3.2-11)$$

以上两式中，按连续傅里叶变换推论，(s, t) 称为空间频率。在上述定义中两个式子前面均有相同的因子 $\frac{1}{\sqrt{MN}}$，这是为了使两个式子更相像。在有的定义中，这两个因子是 1 和 $\frac{1}{MN}$ 或其他形式，这并不影响问题的本质。类似于 3.2.1 小节中的分析，由(3.2-11)式可知，数字图像 $f(m, n)$ 是由 $M\times N$ 个余弦(正弦)基图像线性组合而成，加权因子为 $F(s, t)$，$F(s, t)$ 称为空间频谱。

由于傅里叶变换核是可分离的，式(3.2-10)和(3.2-11)可改写为：

$$F(s, t)=\frac{1}{\sqrt{N}}\sum_{n=0}^{N-1}\left[\frac{1}{\sqrt{M}}\sum_{m=0}^{M-1}f(m, n)\exp\left(-\mathrm{j}2\pi\frac{sm}{M}\right)\right]\exp\left(-\mathrm{j}2\pi\frac{tn}{N}\right) \quad (3.2-12)$$

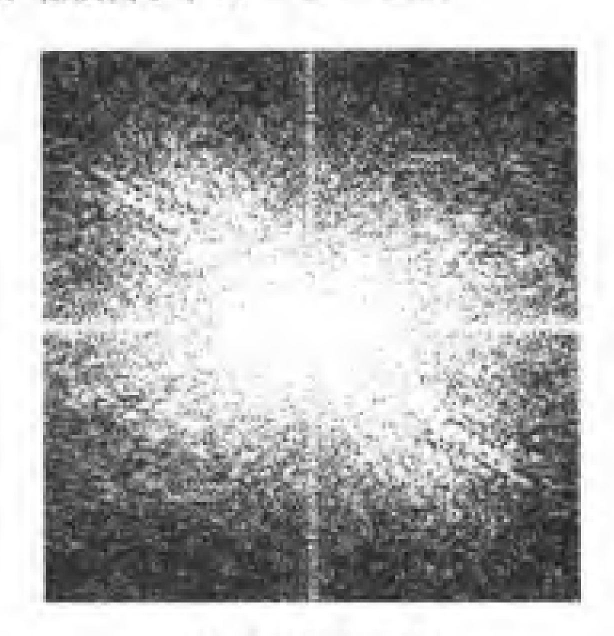
$$f(m, n)=\frac{1}{\sqrt{N}}\sum_{t=0}^{N-1}\left[\frac{1}{\sqrt{M}}\sum_{s=0}^{M-1}F(s, t)\exp\left(\mathrm{j}2\pi\frac{sm}{M}\right)\right]\exp\left(\mathrm{j}2\pi\frac{tn}{N}\right) \quad (3.2-13)$$

以上两式方括号中是一维离散傅里叶变换，方括号外是另一次一维变换，这表明可由两次一维傅里叶变换来实现二维傅里叶变换。其次，正反变换表达式有严格的

对应关系,很相像。因此在计算机中可以用一个一维离散傅里叶变换子程序来完成正、反变换,仅需按要求改动几个参数而已。

图 3.4 是一个二维傅里叶变换的实例,其中图(b)是空间频谱的幅度谱 $|F(s, t)|$, $F(0, 0)$ 位于正方形的中央。在该例子中 $M=N=256$。







(a) 原图　　　　(b) 空间频谱(幅值)

图 3.4　二维傅里叶变换实例

下面讨论以矩阵形式表达的离散傅里叶正、反变换。为书写方便,记:

$$\begin{cases} W_M=\exp\left(-\mathrm{j}\dfrac{2\pi}{M}\right) \\ W_N=\exp\left(-\mathrm{j}\dfrac{2\pi}{N}\right) \end{cases} \tag{3.2-14}$$

记图像矩阵 $[f(m, n)]$ 为 $[\boldsymbol{f}]$,空间频率矩阵 $[F(s, t)]$ 为 $[\boldsymbol{F}]$,则二维离散傅里叶变换可表示成:

$$[\boldsymbol{F}]=[\boldsymbol{U}]^{\mathrm{T}}[\boldsymbol{f}][\boldsymbol{V}] \tag{3.2-15a}$$

式中:

$$[\boldsymbol{F}]=\begin{bmatrix} F(0, 0) & F(0, 1) & \cdots & F(0, N-1) \\ F(1, 0) & F(1, 1) & \cdots & F(1, N-1) \\ \vdots & \vdots & & \vdots \\ F(M-1, 0) & F(M-1, 1) & \cdots & F(M-1, N-1) \end{bmatrix}_{M\times N} \tag{3.2-16a}$$

$$[f]=\begin{bmatrix} f(0,0) & f(0,1) & \cdots & f(0,N-1) \\ f(1,0) & f(1,1) & \cdots & f(1,N-1) \\ \vdots & \vdots & & \vdots \\ f(M-1,0) & f(M-1,1) & \cdots & f(M-1,N-1) \end{bmatrix}_{M\times N} \tag{3.2-16b}$$

$$[\boldsymbol{U}]^{\mathrm{T}}=\frac{1}{\sqrt{M}}\begin{bmatrix} 1 & 1 & 1 & \cdots & 1 \\ 1 & W_M^1 & W_M^2 & \cdots & W_M^{(M-1)} \\ 1 & W_M^2 & W_M^4 & \cdots & W_M^{2(M-1)} \\ \vdots & \vdots & \vdots & & \vdots \\ 1 & W_M^{(M-1)} & W_M^{2(M-1)} & \cdots & W_M^{(M-1)(M-1)} \end{bmatrix}_{M\times M} \tag{3.2-16c}$$

$$[\boldsymbol{V}]=\frac{1}{\sqrt{N}}\begin{bmatrix} 1 & 1 & 1 & \cdots & 1 \\ 1 & W_N^1 & W_N^2 & \cdots & W_N^{(N-1)} \\ 1 & W_N^2 & W_N^4 & \cdots & W_N^{2(N-1)} \\ \vdots & \vdots & \vdots & & \vdots \\ 1 & W_N^{(N-1)} & W_N^{2(N-1)} & \cdots & W_N^{(N-1)(N-1)} \end{bmatrix}_{N\times N} \tag{3.2-16d}$$

$[\boldsymbol{U}]$和$[\boldsymbol{V}]$的大小可以不一样，但它们都是方阵，且都是酉矩阵，即有$[\boldsymbol{U}][\boldsymbol{U}]^{*\mathrm{T}}=\boldsymbol{I}_M$，$[\boldsymbol{V}][\boldsymbol{V}]^{*\mathrm{T}}=\boldsymbol{I}_N$。从式(3.2-16)可知$[\boldsymbol{U}]$和$[\boldsymbol{V}]$是对称阵，故进一步可得到

$$\begin{cases}[\boldsymbol{U}][\boldsymbol{U}]^*=\boldsymbol{I}_M \\ [\boldsymbol{V}][\boldsymbol{V}]^*=\boldsymbol{I}_N\end{cases} \quad 即 \quad \begin{cases}[\boldsymbol{U}]^{-1}=[\boldsymbol{U}]^* \\ [\boldsymbol{V}]^{-1}=[\boldsymbol{V}]^*\end{cases} \tag{3.2-17}$$

由式(3.2-17)可写出矩阵形式的二维离散傅里叶反变换为：

$$[f]=[\boldsymbol{U}]^*[\boldsymbol{F}][\boldsymbol{V}]^* \tag{3.2-18}$$

式中：

$$[\boldsymbol{U}]^*=\frac{1}{\sqrt{M}}\begin{bmatrix} 1 & 1 & 1 & \cdots & 1 \\ 1 & W_M^{-1} & W_M^{-2} & \cdots & W_M^{-(M-1)} \\ 1 & W_M^{-2} & W_M^{-4} & \cdots & W_M^{-2(M-1)} \\ \vdots & \vdots & \vdots & & \vdots \\ 1 & W_M^{-(M-1)} & W_M^{-2(M-1)} & \cdots & W_M^{-(M-1)(M-1)} \end{bmatrix}_{M\times M} \tag{3.2-19a}$$

$$[\boldsymbol{V}]^*=\frac{1}{\sqrt{N}}\begin{bmatrix} 1 & 1 & 1 & \cdots & 1 \\ 1 & W_N^{-1} & W_N^{-2} & \cdots & W_N^{-(N-1)} \\ 1 & W_N^{-2} & W_N^{-4} & \cdots & W_N^{-2(N-1)} \\ \vdots & \vdots & \vdots & & \vdots \\ 1 & W_N^{-(N-1)} & W_N^{-2(N-1)} & \cdots & W_N^{-(N-1)(N-1)} \end{bmatrix}_{N\times N} \tag{3.2-19b}$$

正变换式可进一步改写为

$$[\boldsymbol{F}]=[\boldsymbol{U}][\boldsymbol{f}][\boldsymbol{V}] \tag{3.2-15b}$$

根据(3.1-5)和(3.1-35)式的对应关系，并考虑到在傅里叶变换中，$\boldsymbol{V}$是对称阵，式(3.2-15b)可以写成

$$\boldsymbol{F}=([\boldsymbol{U}]\otimes[\boldsymbol{V}])\boldsymbol{f}=[\boldsymbol{W}]\boldsymbol{f} \tag{3.2-20}$$

式中：$\boldsymbol{F}$应理解为频谱阵列$[\boldsymbol{F}]$按行堆叠得到的一个向量；$[\boldsymbol{W}]$是一个$MN\times MN$的方阵，并可以证明$\boldsymbol{W}$是一个对称酉矩阵。

利用式(3.1-32)和(3.1-33)有：

$$[\boldsymbol{W}]^{\mathrm{T}}=([\boldsymbol{U}]\otimes[\boldsymbol{V}])^{\mathrm{T}}=[\boldsymbol{U}]^{\mathrm{T}}\otimes[\boldsymbol{V}]^{\mathrm{T}}=[\boldsymbol{U}]\otimes[\boldsymbol{V}]=[\boldsymbol{W}]$$

$$\begin{aligned}[\boldsymbol{W}]^{-1}&=([\boldsymbol{U}]\otimes[\boldsymbol{V}])^{-1}=[\boldsymbol{U}]^{-1}\otimes[\boldsymbol{V}]^{-1}=[\boldsymbol{U}]^{*\mathrm{T}}\otimes[\boldsymbol{V}]^{*\mathrm{T}}\\&=([\boldsymbol{U}]^{*}\otimes[\boldsymbol{V}]^{*})^{\mathrm{T}}=[\boldsymbol{W}]^{*\mathrm{T}}\end{aligned}$$

经堆叠之后可以直接应用向量分析的方法。

2. 二维DFT的循环性与对称性

傅里叶变换有许多有趣而重要的性质。下面讨论在二维离散傅里叶变换中与空间频率谱的分布及变换数据量密切相关的两个重要性质：循环性与对称性。

设k, l为整数，把$s+kM$、$t+lN$取代式(3.2-10)中的s、t，可得

$$\begin{aligned}F(s+kM,\ t+lN)&=\frac{1}{\sqrt{MN}}\sum_{m=0}^{M-1}\sum_{n=0}^{N-1}f(m,\ n)\exp\left[-\mathrm{j}2\pi\left(\frac{sm}{M}+\frac{tn}{N}+km+ln\right)\right]\\&=\frac{1}{\sqrt{MN}}\sum_{m=0}^{M-1}\sum_{n=0}^{N-1}f(m,\ n)\exp\left[-\mathrm{j}2\pi\left(\frac{sm}{M}+\frac{tn}{N}\right)\right]\\&=F(s,\ t)\end{aligned} \tag{3.2-21}$$

即
$$F(s+kM,\ t+lN)=F(s,\ t)$$

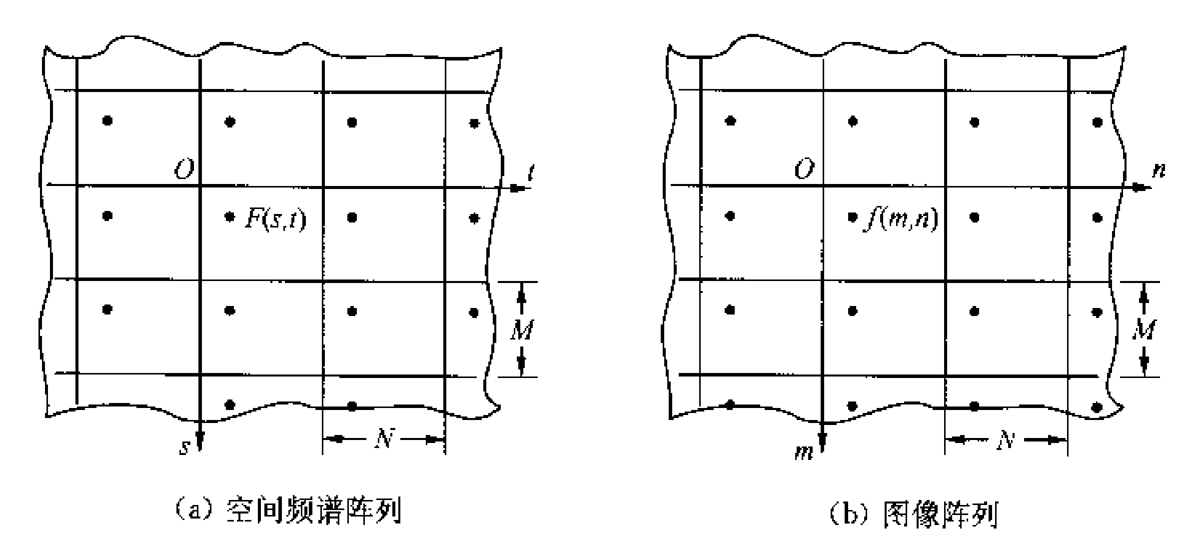
上式表明，在空间频率域上，数据，即$F(s,\ t)$呈现了水平和垂直两个方向上的循环性，如图3.5(a)所示。类似地还可以得到

$$f(m+kM,\ n+lN)=f(m,\ n) \tag{3.2-22}$$

即从傅里叶变换的角度来看，由反变换得到的图像阵列也呈现二维的循环性，如图3.5(b)所示。

由式(3.2-10)可知，M行N列图像矩阵有$M\times N$实数数据，它可以看成$M\times N$个基本图像线性组合的结果，在M、N确定后这些基本图像就已确定；同时得到了$M\times N$各加权因子$F(s,\ t)$，通常$F(s,\ t)$是复数，故实际上得到了$2\times M$

$\times N$ 个实数数据。是不是说数据量不仅没有压缩，反而扩大了一倍呢？由对称性可知，事实并不是这样。



(a) 空间频谱阵列　　　　(b) 图像阵列

图 3.5　空间频谱阵列和图像阵列的循环性

设 k、l 为整数，把 $kM-s$，$lN-t$ 取代(3.2-10)式中的 s，t，可得

$$\begin{aligned} F(kM-s,\ lN-t) &= \frac{1}{\sqrt{MN}}\sum_{m=0}^{M-1}\sum_{n=0}^{N-1} f(m,\ n)\exp\left[-\mathrm{j}2\pi\left(-\frac{sm}{M}-\frac{tn}{N}+km+ln\right)\right] \\ &= \left\{\frac{1}{\sqrt{MN}}\sum_{m=0}^{M-1}\sum_{n=0}^{N-1} f(m,\ n)\exp\left[-\mathrm{j}2\pi\left(\frac{sm}{M}+\frac{tn}{N}\right)\right]\right\}^* \\ &= F^*(s,\ t) \end{aligned} \tag{3.2-23}$$

即

$$F(s,\ t) = F^*(kM-s,\ lN-t)$$

下面的一个具体例子说明上式的含义，令 $k=l=1$，并取 $M=N=256$，可以得到

$$F(s,\ t) = F^*(256-s,\ 256-t) \tag{3.2-24}$$

上式反映了在一个空间频谱循环周期中围绕 $s=\dfrac{M}{2}=128$ 或 $t=\dfrac{N}{2}=128$ 这些特定坐标值而出现的共轭对称性质。

按 $M=N=256$ 作出的空间频谱阵列如图 3.6 所示。正方形 $ABDC$ 被划分成十六块，由(3.2-24)式在图中由箭头指明了彼此具有共轭对称性的对应区域共六对。因此在正方形 $ABDC$ 中仅有打阴影线的十块区域的数据是独立的。考虑到 $F(0,\ 0)$、$F(0,\ 128)$、$F(128,\ 0)$、$F(128,\ 128)$四个值是实数，故阴影区中的复数数据量(按实数计)正好为 $\frac{1}{2}(2\times M\times N)=M\times N$。这一结果表明，经过傅里叶变换后，实数数据量没有增加也没有减少。

通常根据式(3.2-21)表明的循环性把 S_1，S_2，S_3 三块区域移到 S_1'，S_2'，S_3'

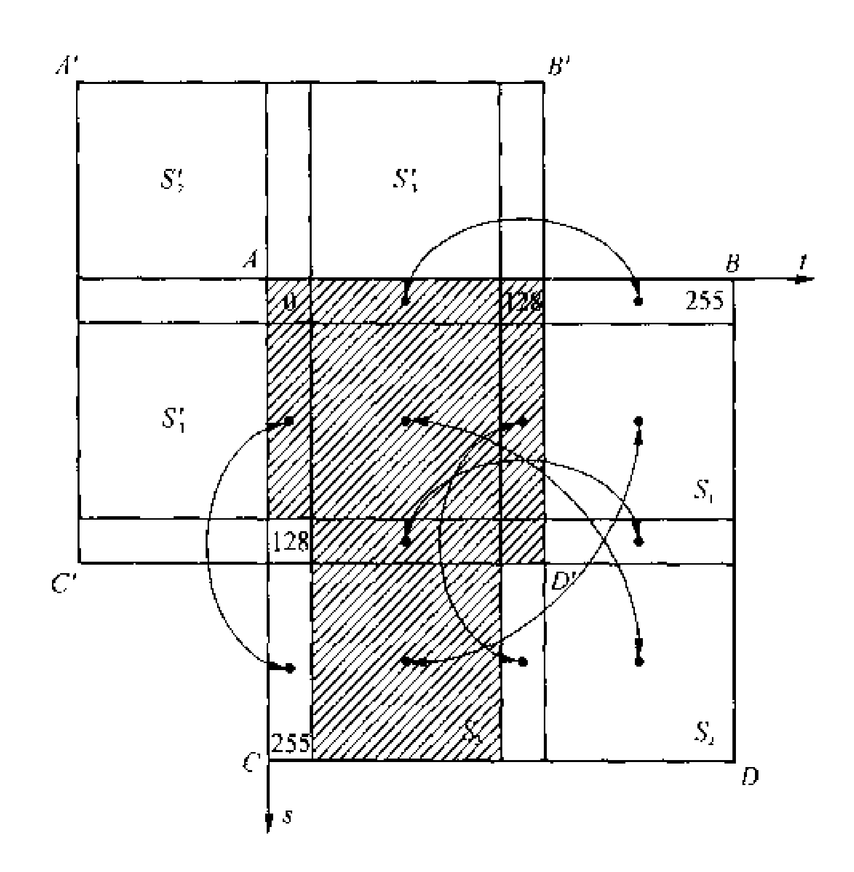


图 3.6 空间频谱阵列的对称性

的位置上去，使得“直流”分量 $F(0, 0)$ 位于正方形 $A'B'D'C'$ 的中央。例如在图 3.4(b)中，就已经利用循环性把 $F(0, 0)$ 点移到正方形中央，$F(0, 0)$ 对应了图像灰度的平均值。

由上述分析可知，对 M 行 N 列图像进行傅里叶变换后，高空间频率成分将分布在 $s=\dfrac{M}{2}$，$t=\dfrac{N}{2}$ 点的附近，即正方形 $ABDC$ 的中央，或正方形 $A'B'D'C'$ 的四个角附近。

进行傅里叶变换之后可以进行一些谱分析，如寻找某特定的空间频率成分进行滤波处理，进行低通、高通、带通、带阻滤波等等。虽然变换前后实数数据量没有变化，但信号能量的分布相对集中，对大多数图像仅用 10%～30%的数据量作反变换得到图像[f]，仅引入较小的误差，一般对主观视觉效果仅有很小的影响。

3.3 余弦变换

在一维傅里叶级数展开式中，如果被展开的函数是实偶函数(即对称于原点)，那么其傅里叶级数中只有余弦函数，它的傅里叶频谱是实数。根据这一特点可以推导出另一种变换——离散余弦变换(DCT)。离散余弦变换在数字图像处理中得

到了有效的应用，在计算机上它还可以利用现成的 FFT 算法得以实现，是值得注意的一种变换手段。本节先讨论一维的情况，再讨论二维 DCT 及其在计算机上的实现。

3.3.1 一维 DCT 表达式的结构

有一离散序列 $f(m)$，$m=0, 1, \cdots, M-1$，如图 3.7(a)所示。

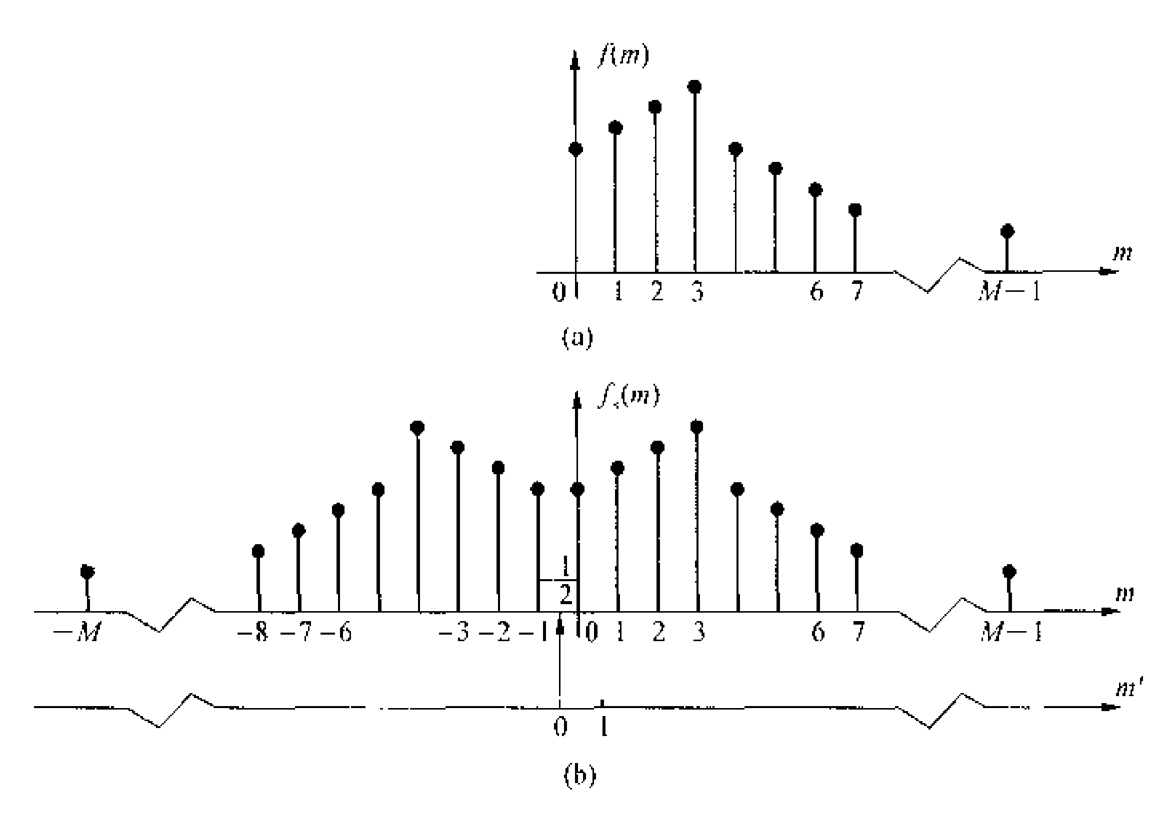


图 3.7　一维 DCT 推导过程简图

现将 $f(m)$ 以 $m=-\frac{1}{2}$ 为对称轴向 $-m$ 方向折叠，得到新的序列 $f_s(m)$。其长度为 $2M$ 点，与 $f(m)$ 的关系为

$$f_s(m)=\begin{cases} f(m), & 0\leqslant m\leqslant M-1 \\ f(-m-1), & -M\leqslant m\leqslant -1 \end{cases} \tag{3.3-1}$$

故 $f_s(m)$ 是关于 $m=-\frac{1}{2}$ 的偶对称序列。

现在把 m' 轴的原点置于 $m=-\frac{1}{2}$ 处，方向与 m 轴相同，则有变换关系：

$$m'=m+\frac{1}{2} \tag{3.3-2}$$

然后在 $m'=-M+\frac{1}{2}$ 到 $m'=M-\frac{1}{2}$ 的范围内进行 $2M$ 点的离散傅里叶变换：

$$F(s)=\frac{1}{\sqrt{2M}}\sum_{m=-M}^{M-1}f_s(m)\exp\left[-\mathrm{j}\,\frac{\pi}{2M}(2m+1)s\right]$$

$$=\frac{1}{\sqrt{2M}}\left(\sum_{m=-M}^{-1}+\sum_{m=0}^{M-1}\right)f_s(m)\exp\left[-\mathrm{j}\,\frac{\pi}{2M}(2m+1)s\right]$$

考虑到式(3.3-1)中的对应关系，上式可进一步写成：

$$F(s)=\frac{1}{\sqrt{2M}}\sum_{m=0}^{M-1}f(m)\exp\left[\mathrm{j}\,\frac{\pi}{2M}(2m+1)s\right]+\frac{1}{\sqrt{2M}}\sum_{m=0}^{M-1}f(m)\exp\left[-\mathrm{j}\,\frac{\pi}{2M}(2m+1)s\right]$$

$$=\sqrt{\frac{2}{M}}\sum_{m=0}^{M-1}f(m)\cos\left[\frac{\pi}{2M}(2m+1)s\right] \tag{3.3-3}$$

上式就是原来的序列 $f(m)$ 的一种变换，变换中仅涉及实数，其中 $s=0,1,2,\cdots,M-1$，变换核

$$g'(s,m)=\sqrt{\frac{2}{M}}\cos\left[\frac{\pi}{2M}(2m+1)s\right] \tag{3.3-4}$$

直接由式(3.3-4)展成的变换矩阵 $[g'(s,m)]$ 为

$$[g'(s,m)]=\sqrt{\frac{2}{M}}\begin{bmatrix} 1 & 1 & \cdots & 1 \\ \cos\frac{\pi}{2M} & \cos\frac{3\pi}{2M} & \cdots & \cos\frac{(2M-1)\pi}{2M} \\ \vdots & \vdots & & \vdots \\ \cos\frac{(M-1)\pi}{2M} & \cos\frac{3(M-1)\pi}{2M} & \cdots & \cos\frac{(2M-1)(M-1)\pi}{2M} \end{bmatrix}$$

其中第一行元素都是 $\sqrt{\frac{2}{M}}$，不符合正交变换矩阵中行向量以及列向量的模应为1的必要条件，为此将第一行元素全部修正为 $\frac{1}{\sqrt{M}}$，得到变换矩阵

$$[\boldsymbol{U}]=\sqrt{\frac{2}{M}}\begin{bmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & \cdots & \frac{1}{\sqrt{2}} \\ \cos\frac{\pi}{2M} & \cos\frac{3\pi}{2M} & \cdots & \cos\frac{(2M-1)\pi}{2M} \\ \vdots & \vdots & & \vdots \\ \cos\frac{(M-1)\pi}{2M} & \cos\frac{3(M-1)\pi}{2M} & \cdots & \cos\frac{(2M-1)(M-1)\pi}{2M} \end{bmatrix} \tag{3.3-5}$$

可以验证上述$[\boldsymbol{U}]$矩阵是一个正交矩阵，但注意它不是一个对称阵。得到矩阵形式表达的一维离散余弦正变换为：

$$\boldsymbol{F} = [\boldsymbol{U}]^{T} \boldsymbol{f} \tag{3.3-6}$$

以求和式表达的 DCT 为：

$$F(s) = c(s) \sqrt{\frac{2}{M}} \sum_{m=0}^{M-1} f(m) \cos\left[\frac{\pi}{2M}(2m+1)s\right] \tag{3.3-7}$$

其中：

$$c(s) = \begin{cases} \dfrac{1}{\sqrt{2}}, & s = 0 \\ 1, & s = 1, 2, \cdots, M-1 \end{cases} \tag{3.3-8}$$

相应的一维离散余弦反变换为：

$$\boldsymbol{f} = [\boldsymbol{U}] \boldsymbol{F} \tag{3.3-9}$$

$$f(m) = \sqrt{\frac{2}{M}} \sum_{s=0}^{M-1} c(s) F(s) \cos\left[\frac{\pi}{2M}(2m+1)s\right] \tag{3.3-10}$$

3.3.2　二维 DCT 表达式的结构

数字图像 $f(m, n)$是一个二维阵列，有两种方法可以使图像对称。第一种方法是使图像沿某条边翻折，如图 3.8(a)，产生一个 $2M \times 2N$ 的图像矩阵，可进行偶对称余弦变换，这种折叠方式与 3.3.1 小节中讨论的情况相对应。第二种方法是使图像翻折时重叠一个像素，如图 3.8(b)，产生一个 $(2M-1) \times (2N-1)$ 的图像矩阵，可进行奇对称余弦变换。下面仅对偶对称余弦变换作一讨论。

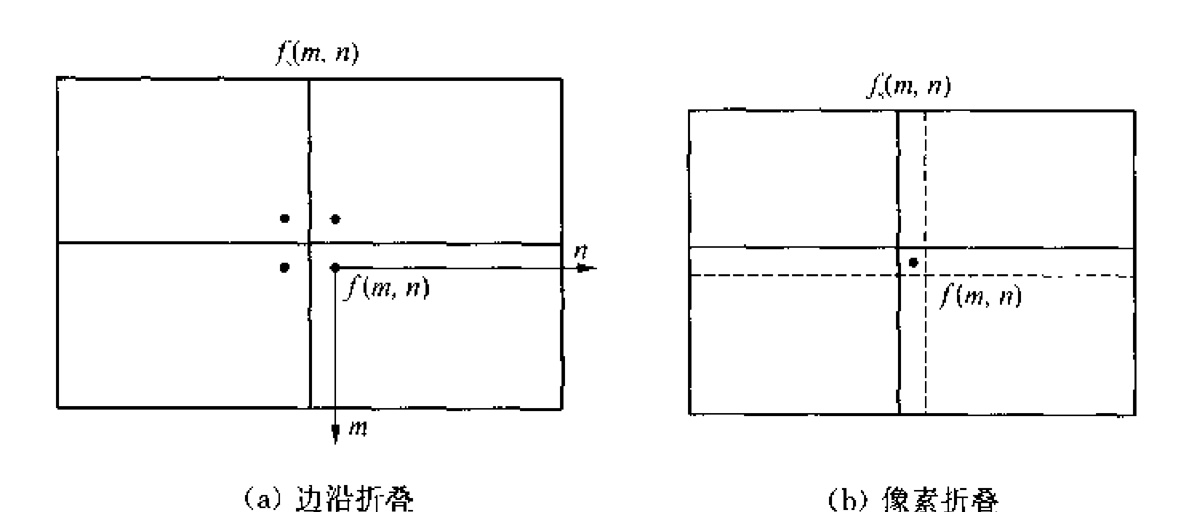
设原图为 $f(m, n)$，$m = 0, 1, \cdots, M-1$，$n = 0, 1, \cdots, N-1$。在偶对称余弦变换时，翻折得到的图像 $f_s(m, n)$是对 $m = -\frac{1}{2}$，$n = -\frac{1}{2}$ 这点对称的。m，n 的取值范围为：

$$m = -M, -M+1, \cdots, -1, 0, 1, \cdots, M-1$$

$$n = -N, -N+1, \cdots, -1, 0, 1, \cdots, N-1$$

图像像素灰度值的对应关系为：

$$f_s(m,n)=\begin{cases} f(m,n), & m\geqslant 0 \text{ 且 } n\geqslant 0 \\ f(-m-1,n), & m<0 \text{ 且 } n\geqslant 0 \\ f(m,-1-n), & m\geqslant 0 \text{ 且 } n<0 \\ f(-1-m,-1-n), & m<0 \text{ 且 } n<0 \end{cases} \tag{3.3-11}$$



(a) 边沿折叠　　(b) 像素折叠

图 3.8　余弦变换对称法

类似于式(3.3-3)，进行以 $m=-\frac{1}{2}$，$n=-\frac{1}{2}$ 为对称点，$2M\times 2N$ 规模的二维傅里叶变换：

$$F(s,t)=\frac{1}{2\sqrt{MN}}\sum_{m=-M}^{M-1}\sum_{n=-N}^{N-1}f_s(m,n)\exp\left\{-j\pi\left[\frac{(2m+1)s}{2M}+\frac{(2n+1)t}{2N}\right]\right\}$$

$$=\frac{1}{2\sqrt{MN}}\left[\sum_{m=-M}^{-1}+\sum_{m=0}^{M-1}\right]\left[\sum_{n=-N}^{-1}+\sum_{n=0}^{N-1}\right]f_s(m,n)$$

$$\exp\left\{-j\pi\left[\frac{(2m+1)s}{2M}+\frac{(2n+1)t}{2N}\right]\right\} \tag{3.3-12}$$

考虑到由式(3.3-11)所反映的 $f_s(m,n)$函数的实对称性质，并把对应项合并，进一步得到

$$F(s,t)=\frac{2}{\sqrt{MN}}\sum_{m=0}^{M-1}\sum_{n=0}^{N-1}f(m,n)\cos\left[\frac{\pi(2m+1)s}{2M}\right]\cos\left[\frac{\pi(2n+1)t}{2N}\right] \tag{3.3-13}$$

在式(3.3-13)中变换核为

$$g(m,n;s,t)=\frac{2}{\sqrt{MN}}\cos\left[\frac{\pi(2m+1)s}{2M}\right]\cos\left[\frac{\pi(2n+1)t}{2N}\right]$$

它是可分离的，故(3.3-13)式可进一步改写为：

$$F(s,t)=\sqrt{\frac{2}{M}}\sum_{m=0}^{M-1}\left\{\sqrt{\frac{2}{N}}\sum_{n=0}^{N-1}f(m,n)\cos\left[\frac{\pi(2n+1)t}{2N}\right]\right\}\times\cos\left[\frac{\pi(2m+1)s}{2M}\right] \tag{3.3-14}$$

以矩阵形式写成的二维离散余弦变换表达式为：

$$[\boldsymbol{F}]=[\boldsymbol{U}]^{\mathrm{T}}[\boldsymbol{f}][\boldsymbol{V}] \tag{3.3-15}$$

式中：$[\boldsymbol{U}]$的结构与(3.3-5)式相同，它是一个 $M\times M$ 阶方阵；$[\boldsymbol{V}]$的结构也与(3.3-5)式类似，但它是一个 $N\times N$ 阶方阵。以(3.3-5)式表达的变换矩阵是正交矩阵，进行了归一化处理。以求和式表达的二维离散余弦正变换式为：

$$F(s,t)=c(s)c(t)\frac{2}{\sqrt{MN}}\sum_{m=0}^{M-1}\sum_{n=0}^{N-1}f(m,n)\cos\left[\frac{\pi(2n+1)t}{2N}\right]\cdot\cos\left[\frac{\pi(2m+1)s}{2M}\right] \tag{3.3-16}$$

式中：函数 $c(*)$的定义见式(3.3-8)。

相应的二维离散余弦反变换式为

$$[\boldsymbol{f}]=[\boldsymbol{U}][\boldsymbol{F}][\boldsymbol{V}]^{\mathrm{T}} \tag{3.3-17}$$

$$f(m,n)=\frac{2}{\sqrt{MN}}\sum_{s=0}^{M-1}\sum_{t=0}^{N-1}c(s)c(t)F(s,t)\cos\left[\frac{\pi(2n+1)t}{2N}\right]\cdot\cos\left[\frac{\pi(2m+1)s}{2M}\right] \tag{3.3-18}$$

3.3.3 由 FFT 算法实现 DCT

离散余弦变换的计算可以由 FFT 算法来实现。下面介绍一维情况下由 FFT 算法实现 DCT 的步骤。

先定

$$W=\exp\left[-\mathrm{j}\frac{2\pi}{2M}\right] \tag{3.3-19}$$

与式(3.2-14)比较可知，这是作 $2M$ 点离散傅里叶变换时用的指数因子。

DCT 所用的余弦函数

$$\begin{aligned}\cos\left[\frac{\pi(2m+1)s}{2M}\right]&=\mathrm{Re}\left[\exp\left(-\mathrm{j}\frac{\pi}{2M}s\right)\exp\left(-\mathrm{j}\frac{2\pi}{2M}ms\right)\right]\\&=\mathrm{Re}\left[\exp\left(-\mathrm{j}\frac{\pi}{2M}s\right)W^{ms}\right]\end{aligned} \tag{3.3-20}$$

式中：Re(*)表示求复数实部的运算。

定义一个新的函数 $f'(m)$，它与原来的函数 $f(m)$，($m=0, 1, \cdots, M-1$)的关系是

$$f'(m)=\begin{cases} f(m), & m=0, 1, 2, \cdots, M-1 \\ 0, & m=M, M+1, \cdots, 2M-1 \end{cases} \tag{3.3-21}$$

希望得到的变换是：

$$F'(0)=F(0)=\frac{1}{\sqrt{M}}\sum_{m=0}^{2M-1} f'(m)W^{s \cdot m},\ s=0 \tag{3.3-22a}$$

$$F'(s)=F(s)=\sqrt{\frac{2}{M}}\sum_{m=0}^{2M-1} f'(m)\cos\left[\frac{\pi}{2M}(2m+1)s\right]$$

$$=\sqrt{\frac{2}{M}}\mathrm{Re}\left[\exp\left(-\mathrm{j}\,\frac{\pi}{2M}s\right)\sum_{m=0}^{2M-1} f'(m)W^{ms}\right],\ 1\leqslant s\leqslant M-1 \tag{3.3-22b}$$

当 $M<s\leqslant 2M-1$ 时的 $F'(s)$并不是我们所需要的，可以舍弃它们，即

$$F'(s)=0,\ M<s\leqslant 2M-1 \tag{3.3-22c}$$

由式(3.3-22)可以写成矩阵形式的表达式：

$$[\boldsymbol{F}]'=\frac{1}{\sqrt{M}}\mathrm{Re}\{[\boldsymbol{G}][\boldsymbol{U}][f]'\}$$

如果将此式展开，并考虑到式(3.3-21)，可以有

$$\begin{bmatrix} F(0) \\ F(1) \\ \vdots \\ F(M-1) \\ 0 \\ \vdots \\ 0 \end{bmatrix}=\frac{1}{\sqrt{M}}\mathrm{Re}\begin{bmatrix} 1 & & & & & & 0 \\ & \sqrt{2}e^{-\mathrm{j}\frac{\pi}{2M}} & & & & & \\ & & \ddots & & & & \\ & & & \sqrt{2}e^{-\mathrm{j}\frac{\pi}{2M}(M-1)} & & & \\ & & & & 0 & & \\ & & & & & \ddots & \\ 0 & & & & & & 0 \end{bmatrix}$$

$$\cdot\begin{bmatrix} 1 & 1 & 1 & \cdots & \cdots & \cdots & 1 \\ 1 & W & W^2 & \cdots & \cdots & \cdots & W^{2M-1} \\ \vdots & \vdots & \vdots & & & & \vdots \\ \vdots & \vdots & \vdots & & & & \vdots \\ \vdots & \vdots & \vdots & & & & \vdots \\ \vdots & \vdots & \vdots & & & & \vdots \\ 1 & W^{2M-1} & W^{2(2M-1)} & \cdots & \cdots & \cdots & W^{(2M-1)(2M-1)} \end{bmatrix} \begin{bmatrix} f(0) \\ f(1) \\ \vdots \\ f(M-1) \\ 0 \\ \vdots \\ 0 \end{bmatrix} \tag{3.3-23}$$

上式中[U]阵就是作一维情况下 $2M$ 点 DFT 用的矩阵。

根据式(3.3-23),由 FFT 算法计算 DCT 的步骤如下:

(1) 把 $f(m)$扩展为 $f'(m)$,在 $f'(m)$中后 M 项全为 0。

(2) 对 $f'(m)$进行 $2M$ 点的 FFT 计算,注意 $W=\exp\left[-\mathrm{j}\dfrac{2\pi}{2M}\right]$。

(3) 各项乘上对应的因子,即进行[$\boldsymbol{G}$]阵的左乘运算。

(4) 取实部,并乘因子$\dfrac{1}{\sqrt{M}}$。

(5) 在 $F'(s)$中,前 M 项即为 $F(s)$。

3.2 节和本节介绍了两种正弦型变换,实际上存在着一系列的正弦型变换[1],可以对应各种偶数、奇数点和奇(正弦)、偶(余弦)的延拓。其中以 DCT 为最好,在图像数据压缩中效果最接近于最佳的情况。

在计算 DCT 时要借用 FFT 算法,在进行 M 点的 DCT 时要利用 $2M$ 点的 FFT,故增加了计算量,为了提高计算效率,还有快速 DCT 算法[2]。

3.4 沃尔什-哈达玛变换

本节所要讨论的哈达玛(Hadamard)变换与所谓沃尔什(Walsh)变换本质上是相同的。关于这类变换的其他资料,可参阅参考文献[3]。

在前人成果的基础上沃尔什(J. L. Walsh)于 1923 年提出了一组在[0, 1]上定义的完备、正交的矩形函数,即沃尔什函数。由于其完备正交性,故可用于正交变换。离散沃尔什函数是在[0, 1]区间上对连续沃尔什函数等间隔取样的结果。在图像处理中,取样点数 M 常取 2 的整数次幂,即 $M=2^l$,这时哈达玛变换矩阵与沃尔什变换矩阵仅仅在行(或列)的排列顺序上不同,因此沃尔什变换与哈达玛变换常统称为"沃尔什-哈达玛变换"。

3.4.1 哈达玛编号的沃尔什-哈达玛变换

这类哈达玛变换矩阵称为哈达玛矩阵，可由递推关系得到，它是由+1和−1元素构成的 $M=2^l$ 阶方阵。在 $M=2^l$ 时，形成该矩阵的递推关系为：

$$[\boldsymbol{H}_2]=\frac{1}{\sqrt{2}}\begin{bmatrix}1 & 1\\1 & -1\end{bmatrix} \tag{3.4-1}$$

$$[\boldsymbol{H}_{2^l}]=\frac{1}{\sqrt{2}}\begin{bmatrix}[\boldsymbol{H}_{2^{l-1}}] & [\boldsymbol{H}_{2^{l-1}}]\\ [\boldsymbol{H}_{2^{l-1}}] & -[\boldsymbol{H}_{2^{l-1}}]\end{bmatrix}=[\boldsymbol{H}_2]\otimes[\boldsymbol{H}_{2^{l-1}}] \tag{3.4-2}$$

根据上两式写出的 $M=4$ 和 $M=8$ 的两个变换矩阵为：

$$[\boldsymbol{H}_4]=\frac{1}{2}\begin{bmatrix}1 & 1 & 1 & 1\\1 & -1 & 1 & -1\\1 & 1 & -1 & -1\\1 & -1 & -1 & 1\end{bmatrix}\quad\begin{matrix}\text{变号次数}\\0\\3\\1\\2\end{matrix} \tag{3.4-3}$$

$$[\boldsymbol{H}_8]=\frac{1}{2\sqrt{2}}\begin{bmatrix}1 & 1 & 1 & 1 & 1 & 1 & 1 & 1\\1 & -1 & 1 & -1 & 1 & -1 & 1 & -1\\1 & 1 & -1 & -1 & 1 & 1 & -1 & -1\\1 & -1 & -1 & 1 & 1 & -1 & -1 & 1\\1 & 1 & 1 & 1 & -1 & -1 & -1 & -1\\1 & -1 & 1 & -1 & -1 & 1 & -1 & 1\\1 & 1 & -1 & -1 & -1 & -1 & 1 & 1\\1 & -1 & -1 & 1 & -1 & 1 & 1 & -1\end{bmatrix}\quad\begin{matrix}\text{变号次数}\\0\\7\\3\\4\\1\\6\\2\\5\end{matrix} \tag{3.4-4}$$

在进行了归一化处理后矩阵 $[\boldsymbol{H}_M]$ 有下述性质：

(1) $[\boldsymbol{H}_M]$ 是一对称阵，即

$$[\boldsymbol{H}_M]^{\mathrm{T}}=[\boldsymbol{H}_M] \tag{3.4-5}$$

(2) $[\boldsymbol{H}_M]$ 是一正交阵，即

$$[\boldsymbol{H}_M]^{\mathrm{T}}[\boldsymbol{H}_M]=[\boldsymbol{H}_M][\boldsymbol{H}_M]^{\mathrm{T}}=\boldsymbol{I}_M \tag{3.4-6}$$

由上两式可得 $[\boldsymbol{H}_M]^{-1}=[\boldsymbol{H}_M]$，这说明哈达玛的正变换矩阵与反变换矩阵完全相

同，即它们的正反变换核相同，进而表明正、反变换的求和表达式也一致。

一维哈达玛正变换核 $h(s, m)$，亦即反变换核是：

$$h(s, m) = \frac{1}{\sqrt{M}}(-1)^{p(s, m)} \quad s, m = 0, 1, \cdots, M-1 \tag{3.4-7}$$

$$p(s, m) = \sum_{i=0}^{l-1} b_i(s) b_i(m) \tag{3.4-8}$$

式中：$b_i(s)$ 表示由二进制数表达整数 s 时，第 i 位的码的值，见表 3-1。

表 3.1　二进制编码

s	$b_3(s)$	$b_2(s)$	$b_1(s)$	$b_0(s)$
14	1	1	1	0
10	1	0	1	0
7	0	1	1	1

在上述规定下，一维哈达玛正、反变换的求和式是相同的，即

$$F(s) = \sum_{m=0}^{M-1} f(m) h(s, m) = \frac{1}{\sqrt{M}} \sum_{m=0}^{M-1} f(m)(-1)^{\sum_{i=0}^{l-1} b_i(s) b_i(m)} \tag{3.4-9}$$

$$f(m) = \sum_{s=0}^{M-1} F(s) h(s, m) = \frac{1}{\sqrt{M}} \sum_{s=0}^{M-1} F(s)(-1)^{\sum_{i=0}^{l-1} b_i(s) b_i(m)} \tag{3.4-10}$$

以矩阵形式写出的哈达玛正、反变换式是

$$\boldsymbol{F} = [\boldsymbol{H}_M]\boldsymbol{f} \tag{3.4-11}$$

$$\boldsymbol{f} = [\boldsymbol{H}_M]\boldsymbol{F} \tag{3.4-12}$$

二维哈达玛的正、反变换核为：

$$h(s, t; m, n) = \frac{1}{\sqrt{MN}}(-1)^{p(s, m)+p(t, n)} \tag{3.4-13}$$

$$p(s, m) + p(t, n) = \sum_{i=0}^{l-1} [b_i(s) b_i(m) + b_i(t) b_i(n)] \tag{3.4-14}$$

显然这个变换核是可分离的，即

$$h(s, t; m, n) = h_1(s, m) h_2(t, n)$$

二维哈达玛正、反变换表达式为

$$F(s, t) = \frac{1}{\sqrt{MN}}\sum_{m=0}^{M-1}\sum_{n=0}^{N-1} f(m, n)(-1)^{\sum_{i=0}^{L-1}[b_i(s)b_i(m)+b_i(t)b_i(n)]} \tag{3.4-15}$$

$$f(m, n) = \frac{1}{\sqrt{MN}}\sum_{s=0}^{M-1}\sum_{t=0}^{N-1} F(s, t)(-1)^{\sum_{i=0}^{L-1}[b_i(s)b_i(m)+b_i(t)b_i(n)]} \tag{3.4-16}$$

由于变换核可分离，又由于$[\boldsymbol{H}_M]$的对称性，二维变换的矩阵形式是

$$[\boldsymbol{F}] = [\boldsymbol{H}_M][\boldsymbol{f}][\boldsymbol{H}_N] \tag{3.4-17}$$

$$[\boldsymbol{f}] = [\boldsymbol{H}_M][\boldsymbol{F}][\boldsymbol{H}_N] \tag{3.4-18}$$

可见正反变换式不仅相同，而且有十分简单的形式。

由式(3.4-1)和式(3.4-2)确定的变换矩阵以及由式(3.4-7)和式(3.4-8)确定的变换核是等价的，用它们进行的哈达玛变换常称为哈达玛编号的沃尔什-哈达玛变换，记作$(\text{WHT})_h$。

3.4.2 列率

在傅里叶变换中用频率一词描述正弦函数组，其过零点在一个区间内是等间隔分布的。频率一词被解释为某个正弦函数在单位时间内所经历的完整的周期数，它正好是函数值过零点次数的一半。

频率的概念可以推广为广义频率，它通常被称为列率。列率被解释为函数在单位时间内其值过零点的平均次数的一半。列率用于描述那些在一个区间内的过零点非等间隔地分布、同时又不一定是周期性的函数。当应用于正弦函数时，列率的定义与频率的定义相一致。

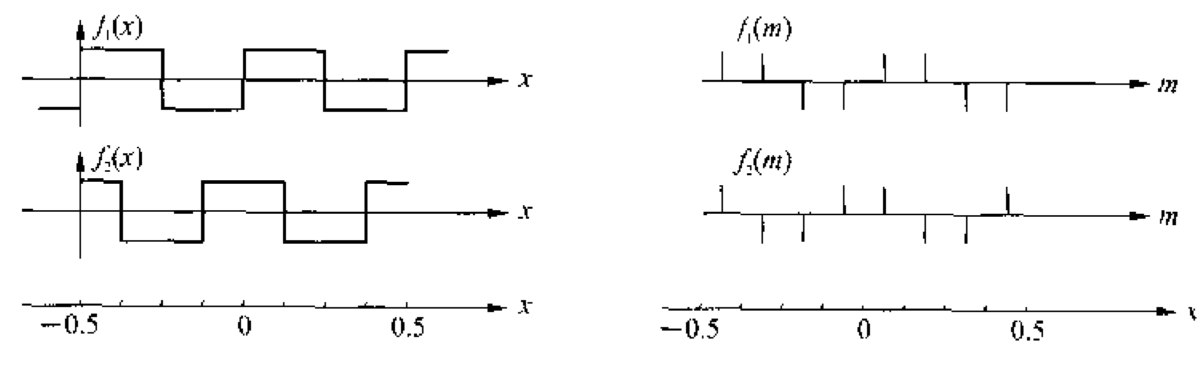


图 3.9 连续函数列率的定义

图 3.10 离散函数列率的定义

例如，图 3.9 示出了在一个半开半闭区间[−0.5, 0.5)上的两个连续函数 $f_1(x)$，$f_2(x)$。由于在该区间上函数有四个过零点，所以列率都等于 2。

对列率的定义稍加修改，便可应用于相应的离散函数 $f(m)$。$f(m)$是对 $f(x)$ 等间隔地取样得到的。如果在单位时间内 $f(m)$的符号变更次数为 k，则列率可定

义为：

$$S=\begin{cases}\dfrac{k}{2}, & k\text{ 为偶数}\\ \dfrac{k+1}{2}, & k\text{ 为奇数}\end{cases} \tag{3.4-19}$$

对图3.9中的 $f_1(x)$ 和 $f_2(x)$ 做等间隔取样，得到两个离散函数 $f_1(m)$ 和 $f_2(m)$，取样点共八个。从图3.10中可知 $k_1=3$，$k_2=4$，故 $f_1(m)$ 和 $f_2(m)$ 的列率也都等于2。

对式(3.4-4)中构成矩阵[$\boldsymbol{H}_8$]的八个离散函数作列率的计算，得到表3.2。

表3.2　[$\boldsymbol{H}_8$]的列率

变换系数 s	0	1	2	3	4	5	6	7
变号次数 k	0	7	3	4	1	6	2	5
列率 S	0	4	2	2	1	3	1	3

3.4.3　沃尔什编号的沃尔什-哈达玛变换

对式(3.4-4)的八个离散函数分析可知，它们的列率 S 不是按自然数的顺序递增的。在某些应用中，往往希望随变换系数的递增，列率也是递增的。具有这种性质的正交变换就是沃尔什编号的沃尔什-哈达玛变换$(\mathrm{WHT})_{\mathrm{W}}$。

以 $M=8$ 为例，相应的变换矩阵[$\boldsymbol{H}'_8$]为：

$$[\boldsymbol{H}'_8]=\frac{1}{2\sqrt{2}}\begin{bmatrix}1&1&1&1&1&1&1&1\\1&1&1&1&-1&-1&-1&-1\\1&1&-1&-1&-1&-1&1&1\\1&1&-1&-1&1&1&-1&-1\\1&-1&-1&1&1&-1&-1&1\\1&-1&-1&1&-1&1&1&-1\\1&-1&1&-1&-1&1&-1&1\\1&-1&1&-1&1&-1&1&-1\end{bmatrix}\begin{matrix}\text{变号次数}\\0\\1\\2\\3\\4\\5\\6\\7\end{matrix} \tag{3.4-20}$$

由此变换系数 s，变号次数 k，列率的关系如表3.3所示。

在进行了归一化处理后，[$\boldsymbol{H}'_M$]阵同样具有式(3.4-5)和(3.4-6)的两个性质，并得到

$$([\boldsymbol{H}'_M])^{-1} = [\boldsymbol{H}'_M]$$

此时一维正反变换核 $h'(s, m)$是

$$h'(s, m) = \frac{1}{\sqrt{M}}(-1)^{q(s, m)}, \quad s, m = 0, 1, \cdots, M-1 \qquad (3.4-21)$$

表 3.3 $[\boldsymbol{H}'_8]$的列率

变换系数 s	0	1	2	3	4	5	6	7
变号次数 k	0	1	2	3	4	5	6	7
列率 S	0	1	1	2	2	3	3	4

$$q(s, m) = \sum_{i=0}^{l-1} r_i(s) b_i(m) \qquad (3.4-22)$$

式中:$b_i(m)$表示由二进制数表达整数 m 时,第 i 位码的值,以符号$\oplus$表示“异或”(模 2 加)运算,$r_i(s)$的含义是

$$\begin{cases} r_0(s) = b_{l-1}(s) \\ r_1(s) = b_{l-1}(s) \oplus b_{l-2}(s) \\ r_2(s) = b_{l-2}(s) \oplus b_{l-3}(s) \\ \vdots \\ r_{l-1}(s) = b_1(s) \oplus b_0(s) \end{cases} \qquad (3.4-23)$$

由上三式作的计算例子见表 3.4。

表 3.4 式(3.4-23)计算实例

m	$b_2(m)$	$b_1(m)$	$b_0(m)$	s	$b_2(s)$	$b_1(s)$	$b_0(s)$	$r_2(s)$	$r_1(s)$	$r_0(s)$	$q(s, m)$	$h'(s, m)$
1	0	0	1	1	0	0	1	1	0	0	0	$\frac{1}{2\sqrt{2}}$
7	1	1	1	7	1	1	1	0	0	1	1	$-\frac{1}{2\sqrt{2}}$
3	0	1	1	6	1	1	0	1	0	1	1	$-\frac{1}{2\sqrt{2}}$

一维正反(WHT)$_W$ 变换的求和式是相同的:

$$F(s) = \sum_{m=0}^{M-1} f(m) h'(s, m) = \frac{1}{\sqrt{M}} \sum_{m=0}^{M-1} f(m)(-1)^{\sum_{i=0}^{l-1} r_i(s) b_i(m)} \qquad (3.4-24)$$

$$f(m)=\sum_{s=0}^{M-1}F(s)h'(s,m)=\frac{1}{\sqrt{M}}\sum_{s=0}^{M-1}F(s)(-1)^{\sum_{i=0}^{l-1}r_i(s)b_i(m)} \tag{3.4-25}$$

以矩阵形式写出的$(WHT)_W$正、反变换式是

$$\boldsymbol{F}=[\boldsymbol{H}'_M]\boldsymbol{f} \tag{3.4-26}$$

$$\boldsymbol{f}=[\boldsymbol{H}'_M]\boldsymbol{F} \tag{3.4-27}$$

二维正反$(WHT)_W$变换核为：

$$h'(s,t;m,n)=\frac{1}{\sqrt{MN}}(-1)^{q(s,m)+q(t,n)} \tag{3.4-28}$$

$$q(s,m)+q(t,n)=\sum_{i=0}^{l-1}[r_i(s)b_i(m)+r_i(t)b_i(n)] \tag{3.4-29}$$

由于$h'(s,t;m,n)$是可分离的，即

$$h'(s,t;m,n)=h'_1(s,m)h'_2(t,n)$$

二维$(WHT)_W$变换的矩阵形式是

$$[\boldsymbol{F}]=[\boldsymbol{H}'_M][\boldsymbol{f}][\boldsymbol{H}'_N] \tag{3.4-30}$$

$$[\boldsymbol{f}]=[\boldsymbol{H}'_M][\boldsymbol{F}][\boldsymbol{H}'_N] \tag{3.4-31}$$

相应的求和式为：

$$F(s,t)=\frac{1}{\sqrt{MN}}\sum_{m=0}^{M-1}\sum_{n=0}^{N-1}f(m,n)(-1)^{\sum_{i=0}^{l-1}[r_i(s)b_i(m)+r_i(t)b_i(n)]} \tag{3.4-32}$$

$$f(m,n)=\frac{1}{\sqrt{MN}}\sum_{s=0}^{M-1}\sum_{t=0}^{N-1}F(s,t)(-1)^{\sum_{i=0}^{l-1}[r_i(s)b_i(m)+r_i(t)b_i(n)]} \tag{3.4-33}$$

3.4.4 快速沃尔什-哈达玛变换

沃尔什-哈达玛变换也可以用快速算法来实现。从以下讨论中可以看出快速沃尔什-哈达玛变换算法的基础也是蝶形运算。由于沃尔什-哈达玛变换的核是可分离的，二维的变换可以由二次一维变换来实现，所以下面只讨论一维的快速算法。

1. 快速哈达玛编号的沃尔什-哈达玛变换

下面以八点的变换来说明快速算法，这时$M=8$。则由式(3.4-11)有

$$\boldsymbol{F}=[\boldsymbol{H}_8]\boldsymbol{f}$$

根据哈达玛矩阵的递归性质，式(3.4－2)可将上式写成

$$\begin{bmatrix} F(0) \\ F(1) \\ F(2) \\ F(3) \\ F(4) \\ F(5) \\ F(6) \\ F(7) \end{bmatrix} = \frac{1}{\sqrt{2}}\begin{bmatrix} [\boldsymbol{H}_4] & [\boldsymbol{H}_4] \\ [\boldsymbol{H}_4] & -[\boldsymbol{H}_4] \end{bmatrix}\begin{bmatrix} f(0) \\ f(1) \\ f(2) \\ f(3) \\ f(4) \\ f(5) \\ f(6) \\ f(7) \end{bmatrix}$$

利用矩阵的分块相乘得

$$\begin{bmatrix} F(0) \\ F(1) \\ F(2) \\ F(3) \end{bmatrix} = \frac{1}{\sqrt{2}}[\boldsymbol{H}_4]\begin{bmatrix} f_1(0) \\ f_1(1) \\ f_1(2) \\ f_1(3) \end{bmatrix} \tag{3.4-34a}$$

$$\begin{bmatrix} F(4) \\ F(5) \\ F(6) \\ F(7) \end{bmatrix} = \frac{1}{\sqrt{2}}[\boldsymbol{H}_4]\begin{bmatrix} f_1(4) \\ f_1(5) \\ f_1(6) \\ f_1(7) \end{bmatrix} \tag{3.4-34b}$$

式中：

$$\begin{aligned} f_1(i) &= f(i) + f(i+4),\ i = 0,\ 1,\ 2,\ 3 \\ f_1(i) &= f(i-4) - f(i),\ i = 4,\ 5,\ 6,\ 7 \end{aligned} \tag{3.4-35}$$

式(3.4－35)可以用蝶形运算来表示，如图3.11中第一次迭代，再利用哈达玛矩阵的递归性及矩阵分块相乘法则对式(3.4－34)进行分解，又得

$$\begin{aligned}
\begin{bmatrix} F(0) \\ F(1) \end{bmatrix} &= \frac{1}{\sqrt{4}}[\boldsymbol{H}_2]\begin{bmatrix} f_1(0) + f_1(2) \\ f_1(1) + f_1(3) \end{bmatrix} = \frac{1}{\sqrt{4}}[\boldsymbol{H}_2]\begin{bmatrix} f_2(0) \\ f_2(1) \end{bmatrix} \\
\begin{bmatrix} F(2) \\ F(3) \end{bmatrix} &= \frac{1}{\sqrt{4}}[\boldsymbol{H}_2]\begin{bmatrix} f_1(0) - f_1(2) \\ f_1(1) - f_1(3) \end{bmatrix} = \frac{1}{\sqrt{4}}[\boldsymbol{H}_2]\begin{bmatrix} f_2(2) \\ f_2(3) \end{bmatrix} \\
\begin{bmatrix} F(4) \\ F(5) \end{bmatrix} &= \frac{1}{\sqrt{4}}[\boldsymbol{H}_2]\begin{bmatrix} f_1(4) + f_1(6) \\ f_1(5) + f_1(7) \end{bmatrix} = \frac{1}{\sqrt{4}}[\boldsymbol{H}_2]\begin{bmatrix} f_2(4) \\ f_2(5) \end{bmatrix} \\
\begin{bmatrix} F(6) \\ F(7) \end{bmatrix} &= \frac{1}{\sqrt{4}}[\boldsymbol{H}_2]\begin{bmatrix} f_1(4) - f_1(6) \\ f_1(5) - f_1(7) \end{bmatrix} = \frac{1}{\sqrt{4}}[\boldsymbol{H}_2]\begin{bmatrix} f_2(6) \\ f_2(7) \end{bmatrix}
\end{aligned} \tag{3.4-36}$$

式中：$f_2(i)$可以用图3.11中第二次迭代完成。由式(3.4-36)最后可得

$$F(0)=\frac{1}{\sqrt{8}}[f_2(0)+f_2(1)]=\frac{1}{\sqrt{8}}f_3(0)$$

$$F(1)=\frac{1}{\sqrt{8}}[f_2(0)-f_2(1)]=\frac{1}{\sqrt{8}}f_3(1)$$

$$F(2)=\frac{1}{\sqrt{8}}[f_2(2)+f_2(3)]=\frac{1}{\sqrt{8}}f_3(2)$$

$$F(3)=\frac{1}{\sqrt{8}}[f_2(2)-f_2(3)]=\frac{1}{\sqrt{8}}f_3(3)$$

$$\vdots$$

$$F(7)=\frac{1}{\sqrt{8}}[f_2(6)-f_2(7)]=\frac{1}{\sqrt{8}}f_3(7) \tag{3.4-37}$$

将这最后一步也用蝶形运算表示，则得图3.11所示的完整的算法流图。

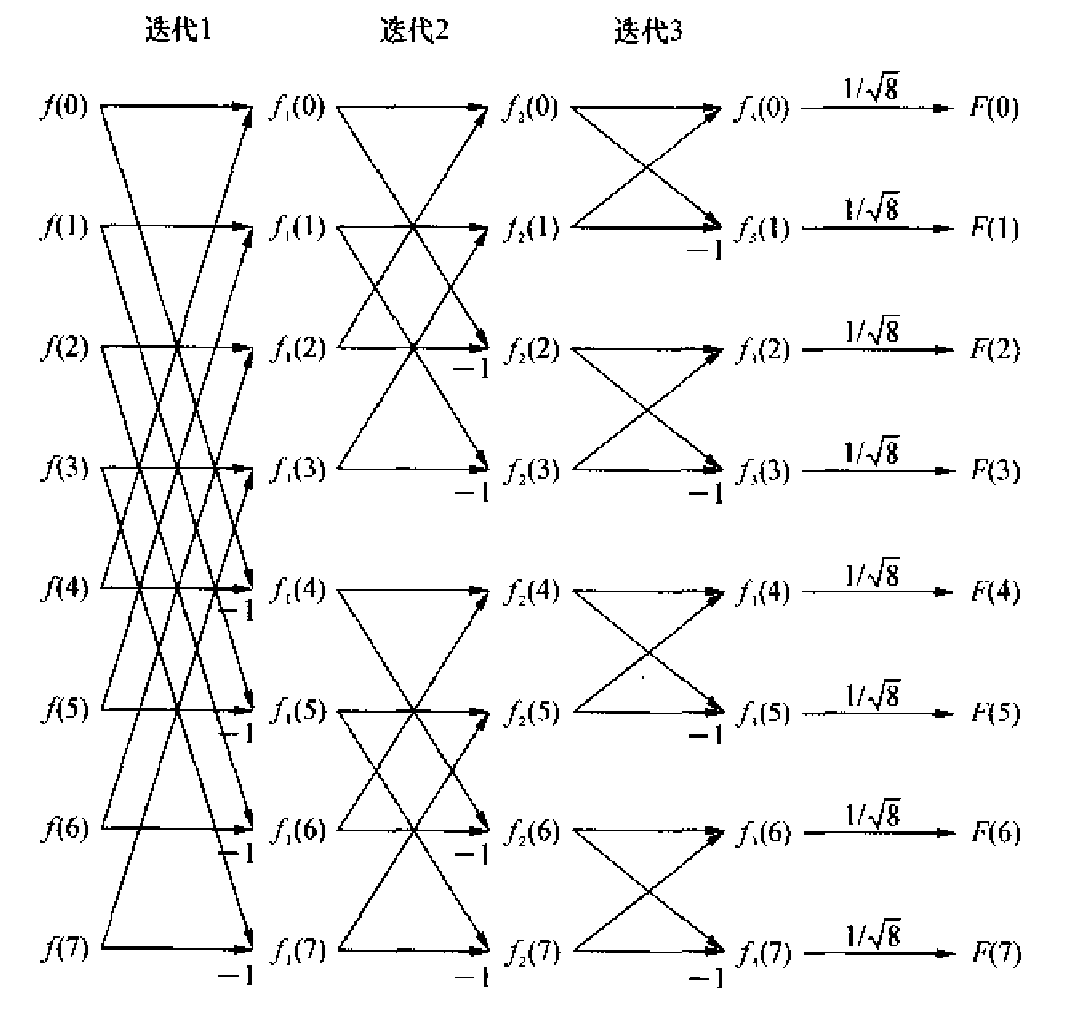


图3.11　快速哈达玛编号的沃尔什-哈达玛变换流图

从算法流图可以看出,快速算法的基本运算是蝶形运算。对于 M 点的变换,每次迭代只需 M 次加法。迭代次数为 $\log_2 M$,所以总的运算次数为 $M\log_2 M$ 次加法。可见所需运算次数大大减少,而且没有复数运算,因而运算速度大大加快。由式(3.4-11)和(3.4-12)可见,反变换和正变换有相同的形式,因此,正变换的快速算法可直接用于反变换,而不必作任何改动。

2. 快速沃尔什编号的沃尔什-哈达玛变换

这种变换的快速算法可以有不同的形式,下面仅介绍其中的一种。因为沃尔什编号的沃尔什-哈达玛变换与哈达玛编号的沃尔什-哈达玛变换的不同仅在变换矩阵中元素的顺序不同,因此这种快速算法只须对上一小节讨论的快速算法稍加改动就可得到。下面介绍修改的方法,仍以 $M=8$ 为例,分下面几步进行。

(1) 将输入序列作比特倒置。

所谓比特倒置就是将输入序列的序号的二进制码反序后作为新的序号。下面举例说明。

序号	0	1	2	3	4	5	6	7
二进制	000	001	010	011	100	101	110	111
反序	000	100	010	110	001	101	011	111
新序	0	4	2	6	1	5	3	7

(2) 定义"反转"。

所谓反转就是把蝶形运算中加法变为减法,减法改为加法。例如式(3.4-37)中

$$f_3(2)=f_2(2)+f_2(3)$$

$$f_3(3)=f_2(2)-f_2(3)$$

反转后

$$f_3(2)=f_2(2)-f_2(3)$$

$$f_3(3)=f_2(2)+f_2(3)$$

(3) 确定所要反转的块。

所谓块就是在算法流图中与上下邻隔开的那些蝶形运算。

反转的块按如下规则确定:第一次迭代内不反转。在其他各次迭代中,将块从上而下以 1, 2, 3, 4, …的次序编号。其中偶数块必须反转。

按以上步骤修改后就得到了沃尔什编号的沃尔什-哈达玛变换。图 3.12 是它

的算法流图。

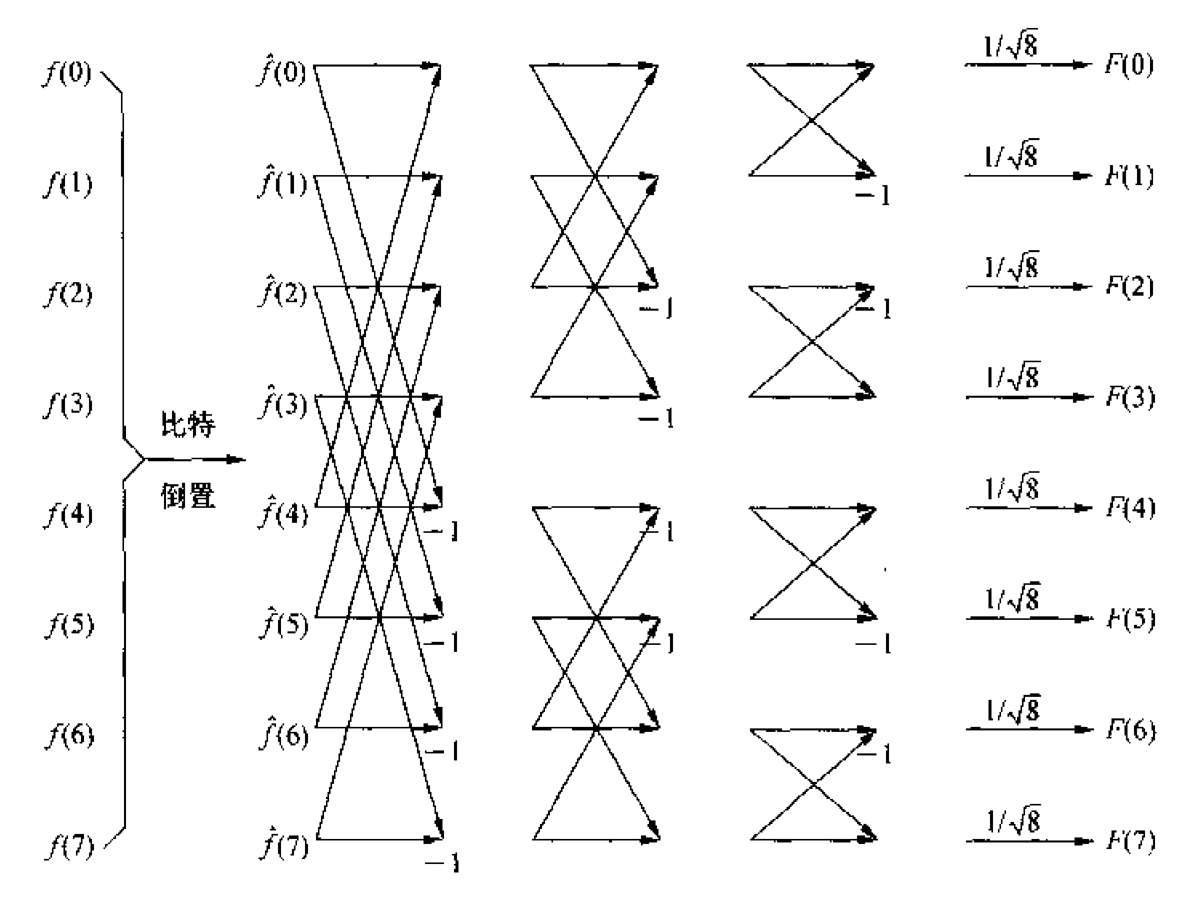


图 3.12 快速沃尔什编号的沃尔什-哈达玛变换流图

3.5 哈尔变换(Haar Transform)

我们将 Haar 变换作为小波变换内容的一个“引子”。其实，哈尔变换是“紧支、二进、正则、归一小波变换”的一个实例。

3.5.1 哈尔变换的基函数(族)

从图 3.13 可以发现：

(1) 除 $h_0(x)$外，基函数的波形都是矩形脉冲对，即均呈 ⊓⊔ 形，只是宽度一尺度不同，位置不同。

(2) 非零区的宽度从 1 以“二进”规律递减为$\frac{1}{2}$，$\frac{1}{4}$，$\frac{1}{8}$，…(限于篇幅，图 3.13 未给出$\frac{1}{8}$以后的波形)。

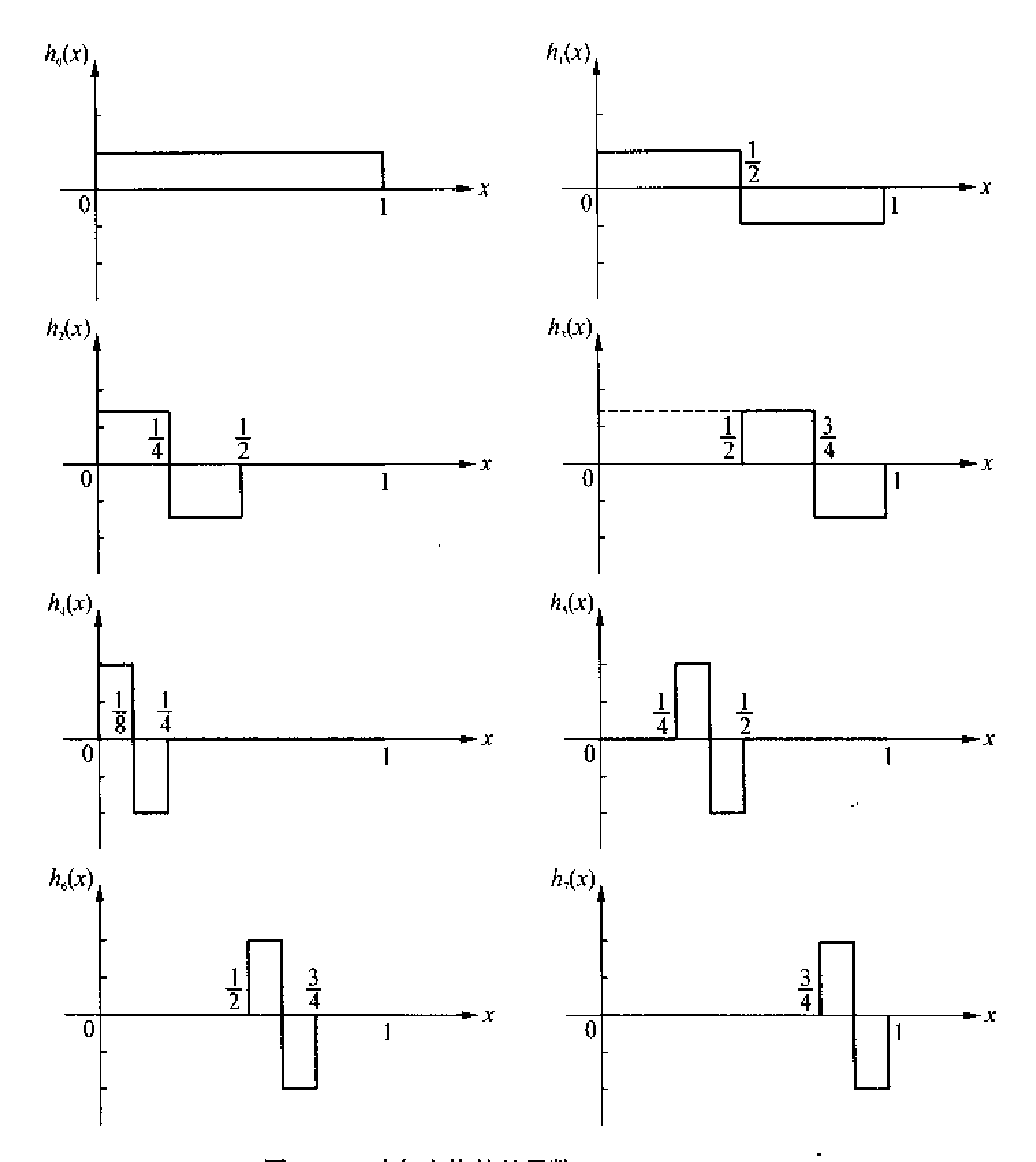


图 3.13 哈尔变换的基函数 $h_k(x)$，$k = 0 \sim 7$

(3) 与前述的 DFT，DCT，KLT 不同，图 3.13 给出的基函数具有以下特性：局部性、尺度伸缩性、平移性，这与小波变换的特性一致。实质上，Haar 变换就是一个小波变换。

3.5.2 离散化的哈尔变换基函数

图 3.14 中，以 $N = 8$ 为例，给出离散化后的哈尔变换基函数的形状。

下面对基函数(族)的形状作出说明：

(1) 共有 N 个基函数(彼此相互正交)。此处，$N = 8$。

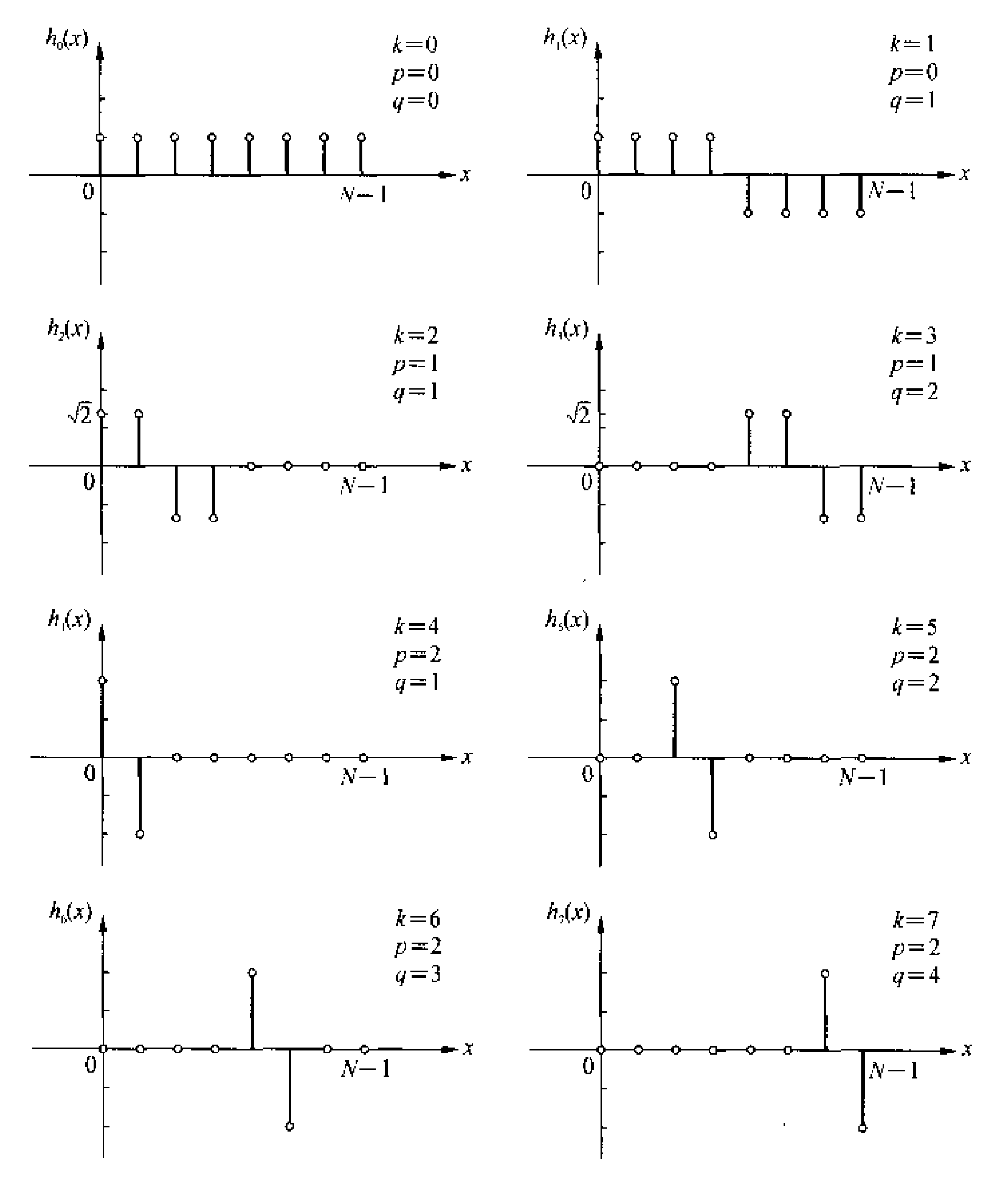


图 3.14　离散化的哈尔变换的基函数 $h_k(x)$，$k=0\sim7$

（2）分为两大类：

$k=0$ 为一大类：对于 x 从 $\frac{0}{N}$，$\frac{1}{N}$，变化至 $\frac{N-1}{N}$，它始终等于 $\frac{1}{\sqrt{N}}$，近似为"直流分量"。

$k=1\sim7$ 为另一大类：有限宽度，其他地方为 0。由正脉冲 ⊓ ，负脉冲 ⊔ 及零组成。

（3）$k=1\sim7$ 这一类可分为三个不同尺度，以 p 表示尺度，有 $p=0, 1, 2$。

$k=1$ 时,4 个正脉冲,4 个负脉冲,在 $0\sim N-1$ 中,没有"零",且 $p=0$;

$k=2,3$ 时,2 个正脉冲,2 个负脉冲,在 $0\sim N-1$ 中,有 4 个"零",且 $p=1$;

$k=4,5,6,7$ 时,1 个正脉冲,1 个负脉冲,在 $0\sim N-1$ 中,有 6 个"零",且 $p=2$。

(4) 同一尺度下的基函数有平移;q 表示平移的位置(平移量)。

尺度越大时,平移步长也越大,以二进制方式增长。以 $\frac{1}{N}$ 为单位步长,$p=1$ 时,具有 4 步长的平移量,即每当 q 增加 1 时,x 的平移量为 $\frac{4}{N}$;$p=2$ 时,具有 2 步长的平移量,即每当 q 增加 1 时,x 的平移量为 $\frac{2}{N}$,此时 $k=4,5,6,7$。两种情形下的平移量都正好覆盖 0~1 区间。

(5) "基函数"的高度则从 $\frac{1}{\sqrt{N}}$ 起,尺度以二进制每"缩小"一档,即宽度除以 2,高度则乘以 $\sqrt{2}$。因此,$k=0,1$ 时,基函数高度为 $\frac{1}{\sqrt{N}}$;$k=2,3$ 时,基函数高度为 $\frac{\sqrt{2}}{\sqrt{N}}$;$k=4,5,6$ 时,基函数高度为 $\frac{2}{\sqrt{N}}$。

(6) 总之,因为 Haar 变换的基函数是有限宽度的,它们在尺度和位置两方面都会有变化。

有了"基函数族",我们就能对信号进行离散变换以及对变换结果作反变换而恢复原始信号。

3.5.3 基函数索引

对上述的一维 Harr 变换情形,Haar 函数需要两个索引 p 和 q 分别表示基函数的"尺度"和"位置"的变化。这和一维 DFT 等变换中只需要一个索引表示频率的情形有明显不同。

也可以用一个数 k 来作索引,用于表示不同基函数的编号。

Haar 函数定义在[0, 1]区间内,将[0, 1]区间等分为 N 份,从 $0\sim\frac{N-1}{N}$。整数 $k(0\leqslant k\leqslant N-1)$ 可以由整数 p, q 唯一确定,具有关系式 $k=2^p+q-1$。这里,p 满足 $2^p\leqslant k$ 且 $2^{p+1}>k$。举例说明,如果 $k=7$,则 $p=2$(因为 $2^2\leqslant 7$ 且 $2^3>7$)。$q-1$ 是 k 被 2^p 减后的剩余量。可对照前一小节 3.5.2 中 $k=0\sim 7$ 时的 p, q 取值来体会。用 p, q 或 k 两种方式对基函数作索引,实质上是等价的。

3.5.4 Haar 函数的定义

前面用图形的表示方式给出了 Haar 函数的描述，本小节回复到函数连续的情形，对 Haar 函数作出数学描述：

$$
\begin{cases}
h_0(x) = \dfrac{1}{\sqrt{N}} \\[2ex]
h_k(x) = \dfrac{1}{\sqrt{N}}\begin{cases}
2^{\frac{p}{2}}, & \dfrac{q-1}{2^p} \leqslant x < \dfrac{q-\frac{1}{2}}{2^p} \quad (\text{正脉冲}) \\[2ex]
-2^{\frac{p}{2}}, & \dfrac{q-\frac{1}{2}}{2^p} \leqslant x < \dfrac{q}{2^p} \quad (\text{负脉冲}) \\[2ex]
0, & \text{其他}
\end{cases}
\end{cases}
\tag{3.5-1}
$$

上式中，x 在[0，1]之间连续变化。对照图 3.13 给出的连续波形图以及图 3.14 所示的 $N=8$ 时的离散波形图的有关说明，连续 Haar 函数的定义式(3.5-1)所表示的不同宽度($p \to p+1$，宽度除以 2)，高度($p \to p+1$，高度乘以$\sqrt{2}$)，位置($p \to p+1$，平移步长除以 2)以及“矩形脉冲对”(对应的高度为$\pm 2^{p/2}$)的含义易于理解。

如果原始信号的波形在某位置处有 形瞬变；由于 Haar 函数为一组正交基，其 Haar 变换结果中相应索引(p，q)的变换系数会很大，其余系数则很小。这样，我们通过 Haar 变换很容易检测出信号的局部变化，而且还能估计出局部变化的尺度大小和位置，如一些线和边的粗细和出现位置。而传统的 DFT 等变换不能这么直观地给出这些信息，信号的位置信息反映在 DFT 变换结果的“相角”中。

3.5.5 Haar 变换的变换矩阵

以 $N=8$ 为例，对照图 3.14 和式(3.5-1)，将每一个基函数作为矩阵的一个行矢量，则 N 个基函数构成一个 $N \times N$ 变换核矩阵：

$$
[\boldsymbol{H}_r] = \frac{1}{\sqrt{8}}\begin{bmatrix}
1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
1 & 1 & 1 & 1 & -1 & -1 & -1 & -1 \\
\sqrt{2} & \sqrt{2} & -\sqrt{2} & -\sqrt{2} & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & \sqrt{2} & \sqrt{2} & -\sqrt{2} & -\sqrt{2} \\
2 & -2 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 2 & -2 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 2 & -2 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 2 & -2
\end{bmatrix}
\tag{3.5-2}
$$

当 N 取最小值 2 时，$[H_2]=\frac{1}{\sqrt{2}}\begin{bmatrix}1 & 1\\1 & -1\end{bmatrix}$。

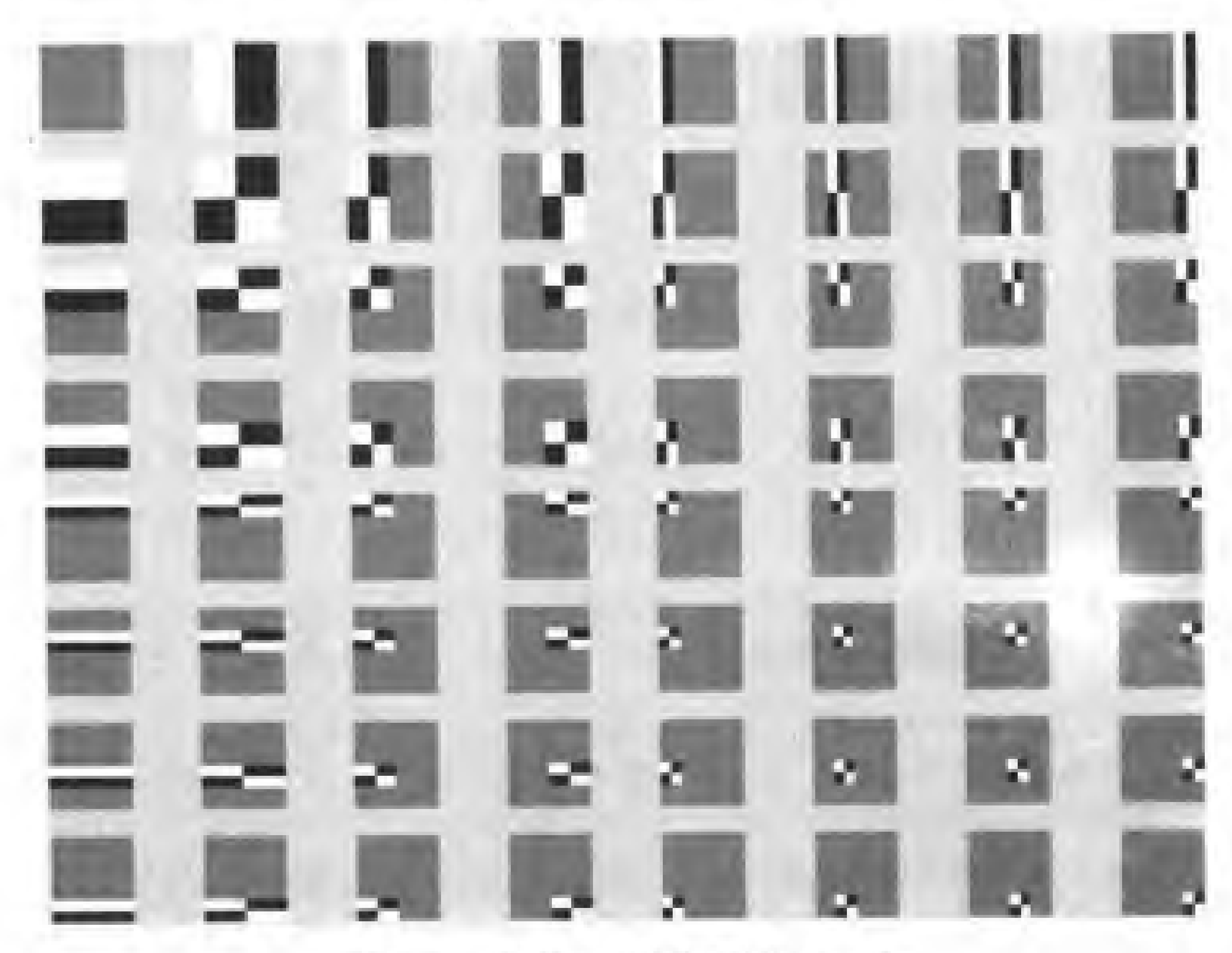


图 3.15 二维 Haar 变换基函数图（$N=8$）

3.5.6 图像的 2D-Haar 变换的变换矩阵

仍以 $N=8$ 为例，那么离散 Haar 变换结果中包含 $8\times8=64$ 个基图像；由于 Haar 变换是以 Haar 函数作为基函数的对称的可分离核的正交变换，所以易于得出这 64 个基函数的形状。

3.6 离散卡-洛变换

离散卡-洛变换（K-L 变换）又称霍特林（Hotelling）变换、特征向量变换或主分量分析。它是以图像的统计性质为基础的。

若 M 行 N 列图像 $f(m,n)$，在信道中传送了 L 次，就收到了以 L 帧图像组成的图像集合 $\{f_1(m,n),f_2(m,n),\cdots,f_L(m,n)\}$。该集合的统计性质与信道的特征

和干扰的性质有关。离散卡-洛变换就是根据图像的统计特性进行的正交变换。

3.6.1 离散卡-洛变换的表达式

在图像集合$\{f_i(m, n)\}$中的每个图像$f_i(m, n)$可以用堆叠方式表达成MN维向量$\boldsymbol{f}_i$：

$$f_i=\begin{bmatrix} f_{i,0} \\ f_{i,1} \\ \vdots \\ f_{i,M-1} \end{bmatrix} \qquad f_{i,j}=\begin{bmatrix} f_i(j, 0) \\ f_i(j, 1) \\ \vdots \\ f_i(j, N-1) \end{bmatrix} \tag{3.6-1}$$

式中：$f_{i,j}$是集合中第i帧图像第j行元素排成的列向量。与式(3.1-19)和(3.1-20)的含义一致。

f向量的协方差矩阵定义为

$$[\boldsymbol{C}_f]=E\{(\boldsymbol{f}-\boldsymbol{m}_f)(\boldsymbol{f}-\boldsymbol{m}_f)^{\mathrm{T}}\} \tag{3.6-2}$$

式中：

$$\boldsymbol{m}_f=E\{\boldsymbol{f}\} \tag{3.6-3}$$

是f的平均值向量，$E\{\cdot\}$表示求统计平均的运算。

在L帧图像组成的集合中，式(3.6-3)和(3.6-2)可由下面两式计算：

$$\boldsymbol{m}_f \approx \frac{1}{L}\sum_{i=1}^{L} f_i \tag{3.6-4}$$

$$\begin{aligned}[\boldsymbol{C}_f] &\approx \frac{1}{L}\sum_{i=1}^{L}(f_i-\boldsymbol{m}_f)(f_i-\boldsymbol{m}_f)^{\mathrm{T}} \\ &= \frac{1}{L}\left[\sum_{i=1}^{L} f_i f_i^{\mathrm{T}}\right]-\boldsymbol{m}_f\boldsymbol{m}_f^{\mathrm{T}}\end{aligned} \tag{3.6-5}$$

上述平均值向量$\boldsymbol{m}_f$是MN维的，而$[\boldsymbol{C}_f]$阵是MN阶方阵。

设$\boldsymbol{a}_i$和λ_i，$i=1, 2, \cdots, MN$，是$[\boldsymbol{C}_f]$的特征向量及其相应的特征值。在排列各λ_i时，使得

$$\lambda_1 \geqslant \lambda_2 \geqslant \cdots \geqslant \lambda_{MN} \tag{3.6-6}$$

而特征向量a_i是MN维向量。因此应当有

$$[\boldsymbol{C}_f]\boldsymbol{a}_i=\lambda_i\boldsymbol{a}_i, \quad i=1, 2, \cdots, MN \tag{3.6-7}$$

f的协方差矩阵$[\boldsymbol{C}_f]$是实对称方阵，一定存在有MN个互为正交的实特征向

量，且各特征值 λ_i 为实数，构成一个 MN 维的完备正交向量系。

对各实特征向量 $\boldsymbol{a}_i$ 进行了归一化处理后，就得到了卡-洛变换的变换矩阵 $[\boldsymbol{A}]$，其第 i 行元素由特征向量 $\boldsymbol{a}_i^{\mathrm{T}}$ 构成，即

$$[\boldsymbol{A}] = \begin{bmatrix} \boldsymbol{a}_1^{\mathrm{T}} \\ \boldsymbol{a}_2^{\mathrm{T}} \\ \vdots \\ \boldsymbol{a}_{MN}^{\mathrm{T}} \end{bmatrix} \tag{3.6-8}$$

且

$$\boldsymbol{a}_i^{\mathrm{T}} \boldsymbol{a}_j = \begin{cases} 1, & i = j \\ 0, & i \neq j \end{cases} \tag{3.6-9}$$

显然，$[\boldsymbol{A}]$是一个 MN 阶正交矩阵。

离散卡-洛变换可表达为：

$$\boldsymbol{g} = [\boldsymbol{A}](\boldsymbol{f} - \boldsymbol{m}_f) \tag{3.6-10}$$

变换后 $\boldsymbol{g}$ 的平均值向量 $\boldsymbol{m}_g$ 为：

$$\begin{aligned} \boldsymbol{m}_g &= E\{\boldsymbol{g}\} = E\{[\boldsymbol{A}](\boldsymbol{f} - \boldsymbol{m}_f)\} \\ &= [\boldsymbol{A}] \cdot E\{\boldsymbol{f}\} - [\boldsymbol{A}]\boldsymbol{m}_f = [\boldsymbol{0}]_{MN\times 1} \end{aligned} \tag{3.6-11}$$

而变换后 $\boldsymbol{g}$ 的协方差矩阵$[\boldsymbol{C}_g]$为：

$$\begin{aligned} [\boldsymbol{C}_g] &= E\{([\boldsymbol{A}]\boldsymbol{f} - [\boldsymbol{A}]\boldsymbol{m}_f)([\boldsymbol{A}]\boldsymbol{f} - [\boldsymbol{A}]\boldsymbol{m}_f)^{\mathrm{T}}\} \\ &= E\{[\boldsymbol{A}](\boldsymbol{f} - \boldsymbol{m}_f)(\boldsymbol{f} - \boldsymbol{m}_f)^{\mathrm{T}}[\boldsymbol{A}]^{\mathrm{T}}\} \\ &= [\boldsymbol{A}]E\{(\boldsymbol{f} - \boldsymbol{m}_f)(\boldsymbol{f} - \boldsymbol{m}_f)^{\mathrm{T}}\}[\boldsymbol{A}]^{\mathrm{T}} \\ &= [\boldsymbol{A}][\boldsymbol{C}_f][\boldsymbol{A}]^{\mathrm{T}} \end{aligned} \tag{3.6-12}$$

根据式(3.6-8)，(3.6-7)，可进一步推导式(3.6-12)：

$$\begin{aligned} [\boldsymbol{C}_g] &= [\boldsymbol{A}][\boldsymbol{C}_f][\boldsymbol{A}]^{\mathrm{T}} \\ &= \begin{bmatrix} \boldsymbol{a}_1^{\mathrm{T}} \\ \vdots \\ \boldsymbol{a}_{MN}^{\mathrm{T}} \end{bmatrix} [\boldsymbol{C}_f][\boldsymbol{a}_1 \cdots \boldsymbol{a}_{MN}] \\ &= \begin{bmatrix} \boldsymbol{a}_1^{\mathrm{T}} \\ \vdots \\ \boldsymbol{a}_{MN}^{\mathrm{T}} \end{bmatrix} [[\boldsymbol{C}_f]\boldsymbol{a}_1,\ \cdots,\ [\boldsymbol{C}_f]\boldsymbol{a}_{MN}] \\ &= \begin{bmatrix} \boldsymbol{a}_1^{\mathrm{T}} \\ \vdots \\ \boldsymbol{a}_{MN}^{\mathrm{T}} \end{bmatrix} [\lambda_1 \boldsymbol{a}_1,\ \cdots,\ \lambda_{MN} \boldsymbol{a}_{MN}] \end{aligned}$$

$$= \begin{bmatrix} a_1^{\mathrm{T}} \\ \vdots \\ a_{MN}^{\mathrm{T}} \end{bmatrix} [a_1 \cdots a_{MN}] \begin{bmatrix} \lambda_1 & & 0 \\ & \ddots & \\ 0 & & \lambda_{MN} \end{bmatrix}$$

最后可得

$$[C_g] = \begin{bmatrix} \lambda_1 & & 0 \\ & \ddots & \\ 0 & & \lambda_{MN} \end{bmatrix} = \Lambda \tag{3.6-13}$$

这表明,经离散卡-洛变换后,g 中的各个元素之间是不相关的;g 中第 i 个元素的方差,就是$[C_f]$的第 i 个特征值λ_i。

与式(3.6-10)对应的离散卡-洛反变换式是

$$f = [\mathbf{A}]^{\mathrm{T}} g + m_f \tag{3.6-14}$$

3.6.2 数据量压缩与误差

离散卡-洛变换在多谱图像的特征选择及图像数据压缩方面有一些应用。

由式(3.6-10)确定的正变换是可逆的,由(3.6-14)式可以完整地得到原图像 f。现在若取前 K 个最大的特征值对应的 K 个特征向量构成新的变换矩阵

$$[A_K] = \begin{bmatrix} a_1^{\mathrm{T}} \\ \vdots \\ a_K^{\mathrm{T}} \end{bmatrix}_{K\times MN} \tag{3.6-15}$$

作一新的变换:

$$g_K = [A_K](f - m_f) \tag{3.6-16}$$

则可由 K 维向量 g_K 代替原来的 MN 维向量 g,在这儿 $K < MN$。

再作反变换,得到了原图 f 的近似值

$$\hat{f} = [A_K]^{\mathrm{T}} g_K + m_f \tag{3.6-17}$$

显然,在这儿 $f \neq \hat{f}$。

为了计算因这一变化而引入的误差,先把 K 维向量 g_K 以补 0 的方法扩充为 MN 维向量

$$g' = \begin{bmatrix} g_K \\ 0 \end{bmatrix}_{MN\times 1}$$

则

$$\hat{f} = [\boldsymbol{A}_K]^{\mathrm{T}} \boldsymbol{g}_K + \boldsymbol{m}_f = [\boldsymbol{A}]^{\mathrm{T}} \boldsymbol{g}' + \boldsymbol{m}_f \tag{3.6-18}$$

引入的均方误差

$$\begin{aligned}\varepsilon &= E\{\| \boldsymbol{f} - \hat{\boldsymbol{f}} \|^2\} = E\{\| [A]^{\mathrm{T}}(\boldsymbol{g} - \boldsymbol{g}') \|^2\} \\ &= E\{[[\boldsymbol{A}]^{\mathrm{T}}(\boldsymbol{g} - \boldsymbol{g}')]^{\mathrm{T}}[[\boldsymbol{A}]^{\mathrm{T}}(\boldsymbol{g} - \boldsymbol{g}')]\} \\ &= E\{(\boldsymbol{g} - \boldsymbol{g}')^{\mathrm{T}}(\boldsymbol{g} - \boldsymbol{g}')\} = E\{\sum_{i=K+1}^{MN} \boldsymbol{g}_i^2\} \\ &= \sum_{i=K+1}^{MN} \lambda_i\end{aligned} \tag{3.6-19}$$

因此在满足式(3.6-6)时，由 g_K 重建的图像$\hat{f}$，引入的误差是最小的。

从最小均方误差的意义上讲，由式(3.6-15)和(3.6-16)所确定的离散卡-洛变换所进行的数据压缩是最优的。这一变换所存在的主要困难是由$[C_f]$来计算变换矩阵$[\boldsymbol{A}]$的计算量较大，同时$[\boldsymbol{A}]$随图像的不同而不同。在一般情况下离散卡-洛变换并无快速算法。参考文献[4]介绍的快速 $K-L$ 变换是在图像为平稳随机过程，且其统计特性满足

$$E[f(m, n)] = 0 \tag{3.6-20a}$$

$$E[f(i, j)f(k, l)] = \rho_1^{|i-k|}\rho_2^{|j-l|} \tag{3.6-20b}$$

的假定下得出的快速算法。算法的实质是当 $f(m, n)$的期望为零，自相关函数满足式(3.6-20b)的时候，离散卡-洛变换成为一种正弦变换，从而可以快速实现之。限于篇幅，对算法的步骤及有关推导在此不进行讨论。

在 3.1.6 小节证明了二维正交变换在变换域中信号能量与原来空间域中的信号能量相等。有

$$\sum_{m=0}^{M-1}\sum_{n=0}^{N-1} |f(m, n)|^2 = \sum_{s=0}^{M-1}\sum_{t=0}^{N-1} |F(s, t)|^2 \tag{3.1-36}$$

成立。但是一般说来在变换域中能量总是比空间域中更为集中，也就是在变换域中对不同的 s, t，$|F(s, t)|$的值相差很大，能量往往集中于变换域中少数几项系数之中。这是因为 s, t 代表空间频率，大的 s, t 值表示图像的细节。这些细节所占的能量是比较小的，忽略一些细节对图像不致发生大的影响。正是由于变换域中存在能量集中现象，因而只要在变换域中保留少数几项系数，丢弃包含能量小的系数，则经过反变换后仍能得到原图的近似图像。只要所丢弃的系数所含的能量很小，丢弃它们所造成的失真不会很大，而这样做可以使数据量得到

明显的压缩。对同一幅图像进行不同种类的变换,在变换域中能量的集中情况不尽相同。因而采用不同种类的变换以后,若都在变换域中只保留能量大的 K 项系数,所造成的失真将随所采用的变换种类的不同而不同。图 3.16 是这种情况的示意图。由图可以看到卡-洛变换在数据压缩方面的性能最好,离散余弦变换略次于卡-洛变换,但差别不大。一般来说 WHT 计算最简单,误差却要大一些。图 3.16 所示结果从概念上看是不难理解的。各种形式的正交变换的反变换就是把变换核所代表的许多基本图像加权相加从而恢复原图。卡-洛变换的基本图像不是固定的,而是随原始图像的不同而变化的。正因为如此,只需要用少数几项就能较好地表示原图,而采用固定基本图像的变换就没有这种有利条件,所以需要较多的项数。但是,从实现的方便性来看,有固定基本图像的变换方式易于实现。

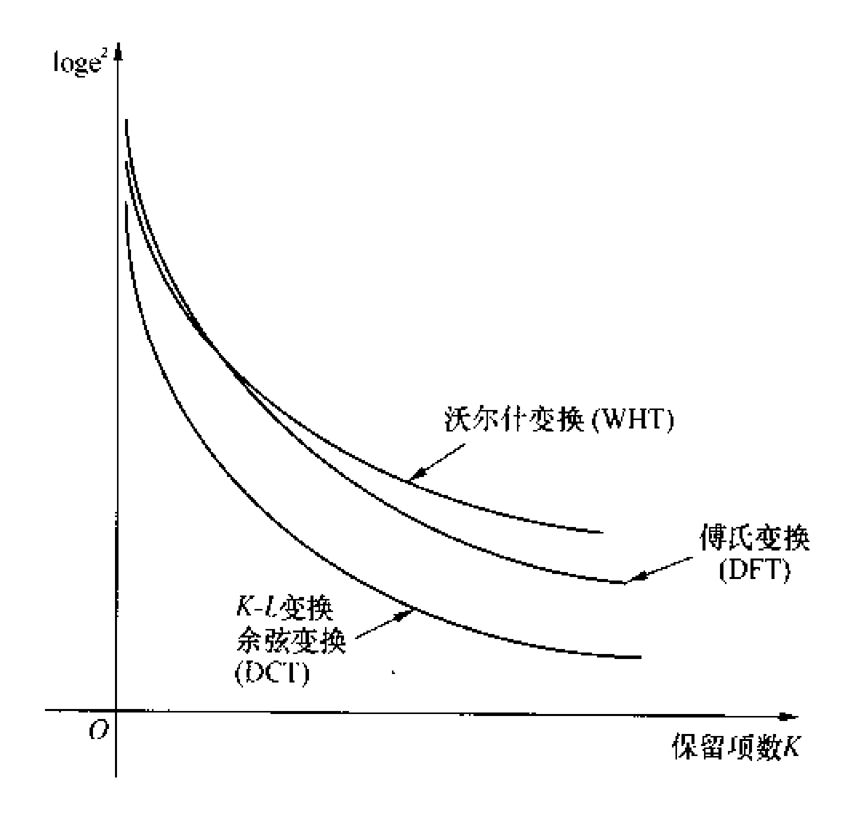


图 3.16 各种变换数据压缩性能的比较

3.7 小波变换和多分辨率分析

近年来小波变换在图像处理领域,尤其是图像压缩、图像增强、边缘检测、特征提取、纹理分析等方面得到了许多应用。

本节主要介绍小波变换的相关知识,同时涉及金字塔分解、子带分解、多

分辨率分析等在概念上与小波变换紧密相关的内容。这是因为对于从事信号处理的专业人士来说，从子带分解来理解 Mallat 的离散小波变换快速算法，比从 Hilbert 空间分解来理解更为直观。而且子带编码本身也是一种很有用的工具。

3.7.1 连续小波变换

1. 波和小波

傅里叶变换以复指数 $e^{+j\omega t}$ 为基函数，它在实数域是正弦波，在 $\pm\infty$ 两个方向上无限扩展，在有的文献中，把它称为“波”。而在 Haar 变换中的基函数是有限宽度的，相对于傅里叶变换中无限宽度的“波”，这样的“有限宽度”的波被称为小波。这种说法并不严格，也未反映问题的实质，只是直观地强调了“可以用有限宽度的或两头衰减很快的基函数来进行变换”。

在 Haar 变换中基函数是不同宽度和位置的。

从 Haar 变换中我们可以看到对于分析瞬态信号（如某位置上的波形，在短时间内非零），用小波变换比用傅里叶变换更直接、直观。因为波形的宽度和位置在小波变换中是通过某一尺度和位置下的小波基与它相匹配来反映的。而在傅里叶变换中，为了在信号为 0 的很大区间里使 $\pm\infty$ 范围内的许多正弦信号“相对消，成为 0”，需要花很大力气，所以结果很不直观。

2. 小波基函数 $\psi(x)$（又称“基本小波”、“母小波”或“基小波”）

设 $\psi(x) \in L^2(\mathbf{R})$，若 $\psi(x)$ 的傅里叶变换 $\Psi(\omega)$ 满足条件 $C_\psi = \int_{-\infty}^{\infty} \frac{|\Psi(\omega)|^2}{|\omega|} d\omega < +\infty$，则称 $\psi(x)$ 为基本小波，或“允许小波”，$C_\psi < \infty$ 这个条件称为允许条件。

由允许条件可知：$\Psi(\omega)$ 在 0 频率处的值等于 0，即 $\psi(x)$ 的直流分量为 0。用数学表达式描述为 $\Psi(0) = 0$，或 $\int_{-\infty}^{\infty} \psi(x)dx = 0$。

基本小波 $\psi(x)$ 有如下性质：

波动性（有正有负，积分值为 0）；

衰减性（$|x| \to \infty$ 时 $\psi(x) \to 0$ 的衰减足够快，即平方可积）；

带通性（在 $\omega = 0$ 和 $\omega \to \infty$ 时 $\Psi(\omega)$ 都应 $\to 0$，否则不满足允许条件 $C_\psi < \infty$）。

3. 连续小波

在选定了具有这样的波动性、衰减性和带通性的基本形状 $\psi(x)$ 以后，把它作一系列的伸缩和平移就可以生成函数族 $\{\psi_{a,b}(x)\}$：

$$\psi_{a,b}(x) = \frac{1}{\sqrt{a}}\psi\left(\frac{x-b}{a}\right) \quad a \in \mathbf{R}_+, \ b \in \mathbf{R} \qquad (3.7-1)$$

式中：a 称为尺度因子；b 称为平移因子；$\psi_{a,b}(x)$被称为连续小波（或称小波函数，小波基函数）。

注意这里 a，b 可以连续变化，是实数而非整数。当 $|a|>1$ 时，小波的波形会"变胖"，$|a|<1$ 时则会"变瘦"。

4. 连续小波变换

连续小波变换是基于这样的一些基本想法：想用一族函数的加权和来表示任意函数 $f(x)$；这一族函数由一系列"小波函数"$\{\psi_{a,b}(x)\}$组成；这些小波函数是由某一个小波基函数平移和伸缩生成的。

函数 $f(x)\in L^2(\mathbf{R})$ 的连续小波变换定义为：

$$W_f(a,b)=\int_{-\infty}^{\infty}f(x)\,\overline{\psi_{a,b}(x)}\,\mathrm{d}x=\langle f(x),\psi_{a,b}(x)\rangle \tag{3.7-2}$$

$\overline{\psi_{a,b}(x)}$为 $\psi_{a,b}(x)$的复共轭，$\langle f(x),\psi_{a,b}(x)\rangle$表示 $f(x)$和 $\psi_{a,b}(x)$的内积。

由连续小波变换 $W_f(a,b)$重构 $f(x)$的反变换为：

$$f(x)=\frac{1}{C_\psi}\int_{-\infty}^{\infty}\int_{-\infty}^{\infty}W_f(a,b)\psi_{a,b}(x)\,\frac{\mathrm{d}a}{a^2}\mathrm{d}b \tag{3.7-3}$$

式中：$C_\psi=\int_{-\infty}^{\infty}\frac{|\Psi(\omega)|^2}{|\omega|}\mathrm{d}\omega$，即"允许条件"中的积分公式。

五点说明：

(1) 因为基小波 $\psi(x)$的均值应为 0，所以各连续小波函数 $\psi_{a,b}(x)$的均值也为 0，在进行连续小波变换时，对 $f(x)$的均值应单独考虑。

(2) 连续小波变换常用内积表达方式 $W_f(a,b)=\langle f(x),\psi_{a,b}(x)\rangle$。

(3) 连续小波变换是冗余的，各个小波基函数 $\psi_{a,b}(x)$之间有很强的相关性，所以它不适合于压缩，图像压缩要用下一小节介绍的离散小波变换。连续小波变换的价值不在于信号的紧凑表示，而在于信号的分解和分析。

一个一维函数 $f(x)$的连续小波变换 $W_f(a,b)$是个双变量函数，变量数多了一个，要求的存储量增加，所以是冗余的。如果连续小波变换用于检测目标，如不同宽度的"线条"，可用$|a|$大的 $\psi_{a,b}(x)$在不同位置 b 处检测粗线条，而用$|a|$小的 $\psi_{a,b}(x)$检测细线条。

(4) 卷积表达和滤波器族分解：用 $\tilde{\psi}_{a,b}(x)$表示 $\psi_{a,b}(x)$的翻转共轭，则连续小波变换为：

$$\begin{aligned}W_f(a,b)&=\int_{-\infty}^{\infty}f(x)\,\overline{\psi_{a,b}(x)}\,\mathrm{d}x\\&=\langle f(x),\psi_{a,b}(x)\rangle=f(x)*\tilde{\psi}_{a,b}(x)\end{aligned} \tag{3.7-4}$$

若将 $\bar{\psi}_{a,b}(x)$看成是某带通滤波器的冲激响应,则连续小波变换 $W_f(a,b)$相当于信号 $f(x)$经过冲激响应为 $\bar{\psi}_{a,b}(x)$的带通滤波器的输出。这就把小波变换和用滤波器族进行信号分解(分解成不同子带的分量)联系起来了。而反变换则是将各滤波器的输出 $W_f(a,b)$经 $\psi_{a,b}(x)$再滤一次,或再作卷积,并作适当伸缩后组合在一起。

(5) 基小波 $\psi(x)$的例子:Haar 小波(正交小波基)

$$\psi(x)=\begin{cases}1, & 0\leqslant x<\frac{1}{2}\\ -1, & \frac{1}{2}\leqslant x<1\\ 0, & 其他\end{cases} \tag{3.7-5}$$

3.7.2 离散小波变换和小波级数

1. 连续小波函数的离散化

将连续小波函数 $\psi_{a,b}(x)=\frac{1}{\sqrt{a}}\psi\left(\frac{x-b}{a}\right)$中的伸缩因子$a$ 和平移因子b 按下列规律离散取样:$a=a_0^{-m}$;$b=nb_0a_0^{-m}$,则得离散小波函数:

$$\psi_{m,n}(x)=a_0^{\frac{m}{2}}\psi(a_0^m x-nb_0) \tag{3.7-6}$$

其中,a、b 离散,x 仍连续变化。小波变换由下式给出:

$$\begin{aligned}W(m,n)&=\int_{-\infty}^{\infty}f(x)\overline{\psi_{m,n}(x)}\mathrm{d}x=a_0^{\frac{m}{2}}\int_{-\infty}^{\infty}f(x)\overline{\psi(a_0^m x-nb_0)}\mathrm{d}x\\&=\langle f,\psi_{m,n}\rangle\end{aligned} \tag{3.7-7}$$

现在考虑反变换,由于a, b 是离散取样,可用整数 m, n 来表示,所以原反变换式中的二重积分变形为求和计算:

$$f(x)=\sum_{m=-\infty}^{\infty}\sum_{n=-\infty}^{\infty}W(m,n)\psi_{m,n}(x) \tag{3.7-8}$$

现在的情况相当于把 $f(x)$进行级数展开,因此上式被称为"小波级数展开"。进一步地,如果 x 也离散化,则有如下的"离散小波变换":

正变换: $$W(m,n)=\sum_{i=-\infty}^{\infty}f(i)\overline{\psi_{m,n}(i)} \tag{3.7-9}$$

反变换: $$f(i)=\sum_{m=-\infty}^{\infty}\sum_{n=-\infty}^{\infty}W(m,n)\psi_{m,n}(i) \tag{3.7-10}$$

2. 二进小波

二进小波是离散小波的一个特例。取 $a_0 = 2$，则 $a = 2^{-m}$，即对二进小波来说，尺度的伸缩只能是“成倍”或“减半”。这是最常见的尺度变换方式，称为“二进伸缩”。此时 $b = nb_0 2^{-m}$，故随着尺度的成倍/减半变化，步长也相应变大/缩小，则有“二进小波”：

$$\psi_{j,k}(x) = 2^{\frac{j}{2}} \psi(2^j x - k) \tag{3.7-11}$$

式中：j 表示尺度；k 表示步长。

二进小波变换由下式给出：

正变换：
$$W(j, k) = \sum_i f(i) \overline{\psi_{j,k}(i)} \tag{3.7-12}$$

反变换：
$$f(i) = \sum_{j=-\infty}^{\infty} \sum_{k=-\infty}^{\infty} W(j, k) \psi_{j,k}(i) \tag{3.7-13}$$

式中 i 表示 x 的离散化。可见二进小波变换和上小节的离散小波变换有相同形式的变换公式，区别在于上小节中 $\psi_{m,n}(i)$ 的尺度不一定是二进变化的。

严格地说，二进小波应当满足稳定性条件，为了得到稳定的重构算法，还要求函数族 $\{\psi_{m,n}\}$ 构成一个“框架”，限于篇幅，这里不再叙述，请参考相关文献。

上面给出了一些名词及含义，并指出了构造小波函数的方法(连续/离散/二进的母小波通过伸缩和平移来形成小波函数族，然后用正/反变换公式来推求 $W(j, k)$ 和从 $W(j, k)$ 重构 $f(i)$。

实际应用中，小波变换及反变换大多数是用 Mallat 的快速算法来实现的。Mallat 算法的实现采用了子带分解的方法，因此我们先从子带讲起，并从金字塔算法着手引入，这些内容本身也有很广泛的应用。

3.7.3 金字塔编码算法

金字塔编码由 Burt 和 Adelson 于 1983 年提出。它把原图分解成许多不同分辨率的子图像，这些子图像垒起来上小下大，形似金字塔，故得名。金字塔算法是图像多分辨率表示方法中的一种，Mallat 的快速小波变换(FWT)方法就是基于多分辨率分析的。

除了作为“引子”引出子带和快速小波变换(FWT)，金字塔表示法本身还有很多用途：

(1) 它是计算机视觉中常用的多分辨率表示法，便于计算机从不同分辨率的图像中分析不同的物体。金字塔的上层便于分析较大的物体特征，如平缓变化的边缘，若要在塔的下层检测到它，则需要很大的滤波窗口。比如，对图像作 5 层金

字塔式的多分辨率分析，在金字塔的顶层 3×3 的小窗即可检测到的特征，在底层则需要 $(2^4\times3)\times(2^4\times3)$ 大小的检测窗。所以从上层看全貌，从下层看细节的分析方式较为理想和方便，并且按照先上层后下层的顺序进行，上层的全貌信息可指导下层的细节分析。

(2) 和人眼视觉特性相似。人眼可以看到“粗/细”不同的物体特征，但人眼对不同空间分辨率(即不同的“空间频率”)的细节敏感度不同。拉普拉斯金字塔编码作信号压缩时，对人眼敏感的上层信息作细量化，而对人眼不敏感的下层信息作粗量化，甚至丢弃最底层信息。金字塔表示法是一种冗余的表示。

1. 高斯金字塔

设原图大小为 $(2^N+1)\times(2^N+1)$，如果不满足，则需要作边缘拓展。把它作为金字塔的底层(第 0 层)G_0，则高斯金字塔通过如下步骤生成：

(1) 把下一层图像 G_l 用具有低通特性的窗函数作滤波(卷积)(设窗函数为 $W(m, n)$。例如可用高斯形冲激响应函数，或截短至 5×5)，得到模糊的大小暂未变小的图。

(2) 把卷积结果作隔行隔列亚抽样，得到$\frac{1}{4}$大小的“上一层”图像 G_{l+1}。因为是对滤出的低频成分作亚抽样，所以通过选择合适的滤波参数可以做到无混迭，不会因为亚抽样而引入附加失真。

(3) 重复(1)、(2)步骤，得到一系列上层图，形成金字塔。迭代分解过程用式子表示为：

$$G_l(i, j)=\sum_{m=-2}^{2}\sum_{n=-2}^{2}W(m, n)G_{l-1}(2i+m, 2j+n) \qquad (3.7-14)$$

引入“缩小算子”Reduce 可以得到简化表示式：

$$G_l=\text{Reduce}(G_{l-1}) \qquad (3.7-15)$$

显然，高斯金字塔是一种有冗余的表示法。在底层 G_0，即原图信息之外，增加了$\frac{1}{3}$的数据量；这些上层信息仅是低频分量，附加上去便于分析而已。

2. 拉普拉斯金字塔[17]

继续上小节的内容，但从金字塔的上层向下考虑。如果有了上一层 G_l，能否反推下一层 G_{l-1} 呢？答案是否定的，因为 G_l 中不含“细节”，想反推必须补充细节。要补充的细节就是拉普拉斯金字塔的某一层。

定义“放大算子”Expand，它对 G_l 进行内插放大，得到放大的图 $G_{l,1}$，即 $G_{l,1}=\text{Expand}(G_l)$，所得到的 $G_{l,1}$ 大小和 G_{l-1} 一样，但所含细节少于 G_{l-1}。若用 L_{l-1} 表示

此细节图，L_{l-1} 的大小和 G_{l-1} 即下层图一样大，则 $G_{l-1}=G_{l,1}+L_{l-1}$，这里 L_{l-1} 即拉普拉斯金字塔的 $l-1$ 层。如此类推，一层层恢复下去，直到底层，可恢复"原图" G_0。所以拉普拉斯金字塔的每一层是高斯金字塔的本层与上一层的"放大图"之间的差异。

类似地，拉普拉斯金字塔也是由大小逐级减半$\left(在二维的情况下，是\frac{1}{4}\right)$的各层 L_0，L_1，…，L_N 所构成，其中 N 为图像分解层数。除最上层外，每层都用"差值图"，即高斯金字塔本层－Expand(高斯金字塔上层)表示，也就是因图像缩小而丢失的细节。用式子表示：

$$L_l=\begin{cases}G_l-\text{Expand}(G_{l+1})=G_l-G_{l+1,1}, & l<N\\ G_N, & l=N\end{cases} \tag{3.7-16}$$

从滤波器的角度看，高斯金字塔每层的频带逐层减少(约等于减半，但不是严格减半)，而拉普拉斯金字塔每层相当于原图经不同的带通滤波器的输出。四层的拉普拉斯金字塔($N=3$)如图3.17所示。

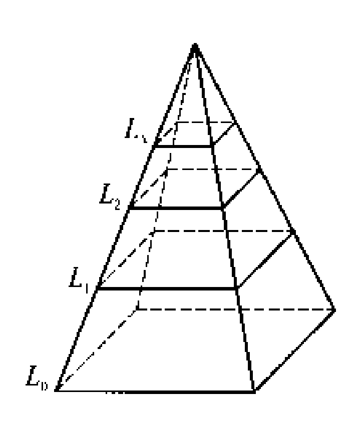


图3.17　拉普拉斯金字塔示意图

由拉普拉斯金字塔 L_0，L_1，…，L_N 恢复原图 G_0 的过程可用递推的 Expand 式子来表示：$G_{l,1}=\text{Expand}(G_l)$，若把 G_l 记作 $G_{l,0}$(G_l 经"0"次放大，即 G_l)，则"放大"两次可记作 $G_{l,2}=\text{Expand}(G_{l,1})=\text{Expand}(\text{Expand}(G_{l,0}))$，依此类推，$G_{l,k}=\text{Expand}(G_{l,k-1})$。类似地，对 L_l 也可同样表示这种"放大"，即 $L_{l,k}=\text{Expand}(L_{l,k-1})$。因此，我们可以推出 $G_0=\sum\limits_{l=0}^{N}L_{l,l}$，即拉普拉斯金字塔每层放大 l 次，放到和原图一样大，然后"总加"，即可恢复原图 G_0。

Expand 是 Reduce 的"逆算子"，对照 Reduce 的"滤波"和亚抽样，现在是"内插"，或补零后低通滤波。$G_{l,1}=\text{Expand}(G_l)$ 可具体化为：

$$G_{l,1}(i,j)=4\sum_{m}^{2}\sum_{2n=-2}^{2}W(m,n)G_l'\left(\frac{i+m}{2},\frac{j+n}{2}\right) \tag{3.7-17}$$

式中：

$$G_l'(p,q)=\begin{cases}G_l(p,q), & p,q\text{ 为整数}\\ 0, & \text{其他}\end{cases}$$

即对上一层图 G_l 先放大，在没有数据处"补零"，然后进行"低通滤波"(空域卷积)，可以就用 Reduce 时的低通冲激响应函数作低通滤波。对拉普拉斯金字塔的每一层，放大算子也一样这样做。

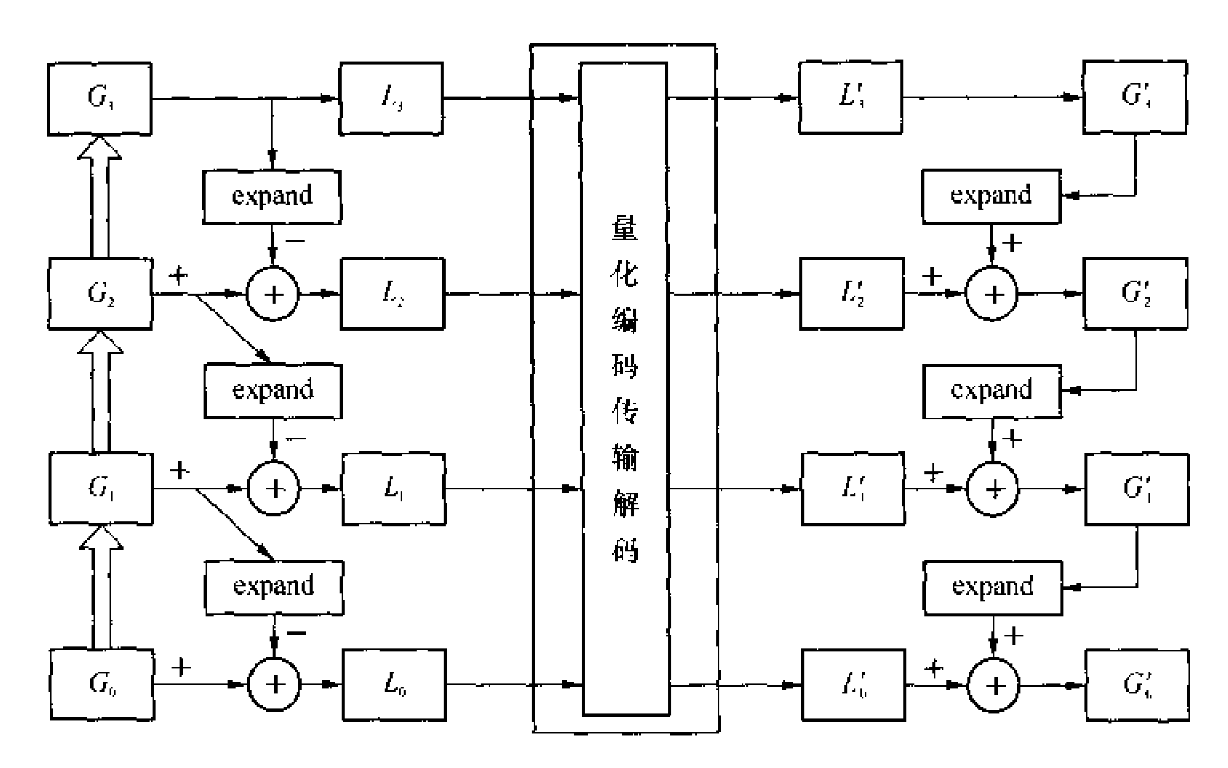


图 3.18 金字塔编码框图

3. 金字塔编码框图

图 3.18 是用金字塔进行压缩编解码的框图。对它说明如下：

(1) 不管怎么具体实现 Reduce，Expand，因为从框图结构上保证了 $G_0=G_1+L_0$，所以只要收发 Expand 做法一致，总可以由 G_1，L_0 恢复 G_0。若 Reduce，Expand 做得较好则层间相关性较小(指 L_0 与 G_1 间)，容易提高压缩比。

(2) 大致上半倍频程式的层次分解(频带从低到高：L_N，L_{N-1}，…，L_1，L_0)。

(3) 从数据压缩的角度看问题

L_i 是差值图，其灰度值的均值约等于 0，方差较小，由率失真定理，它比 G_0 容易编码，即对 G_1，L_0 编码所需比特数可少于直接对 G_0 编码的情形。而且，可以利用人眼对"中间分辨率"细节敏感的特性，对各层设不同量化步长，以达到压缩效果。

(4) 它是一种"冗余"表示$\left(多出\frac{1}{3}数据量\right)$，而下面即将讲述的子带/小波编码无此冗余。

(5) 此框图不反映"方向性"，所以金字塔编码无法利用视觉中"方向性"特性，而子带/小波分解有方向性。

3.7.4 子带编码

金字塔方法提出了一种通过滤波，亚抽样实现多分辨率分析的表示方法，但它有三个缺点：一是一种冗余表示法；二是它未考虑方向性；三是虽然各层信号带宽

大致上以半倍频程方式递减，但这种滤波特性不严格（各层间残余相关性较大）。

子带编码在上述三个方面都比金字塔编码有改进。

1. 子带编码原理

子带编码最早用于语音编码，先用分解滤波器组将信号分成不同频率成分的分量或子带，再将这些分量进行亚抽样，从而实现信号的子带分解。

解码端先补零以恢复原来的采样率，然后用合成滤波器组对各路子带信号进行内插，最后将内插后的各路子带信号相加，恢复原信号。如果分解/合成滤波器组设计得好的话，理论上可以无失真恢复原信号。

一路信号可以分解成多路子带信号，但最感兴趣的是将一路一维信号分解成二路。一维信号的二频带分解、合成框图如图 3.19 所示：

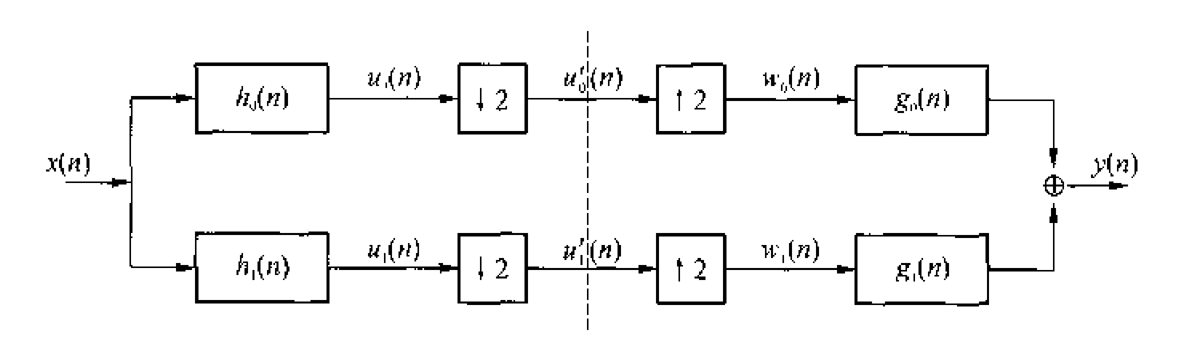


图 3.19　子带分解合成框图

一维信号 $x(n)$ 经分解滤波器组（低通 $h_0(n)$，高通 $h_1(n)$）分解成低频分量 $u_0(n)$ 和高频分量 $u_1(n)$，再经亚抽样（使总数据量不上升），得到二路子带信号；收端在原抽去的样点处“补零”，得到 $w_0(n)$，$w_1(n)$，它们经合成滤波器组 $g_0(n)$，$g_1(n)$ 内插，然后相加，输出复原信号 $y(n)$。在设计得好的情况下可以实现无失真恢复。

图像的二维子带分解框图如图 3.20 所示：

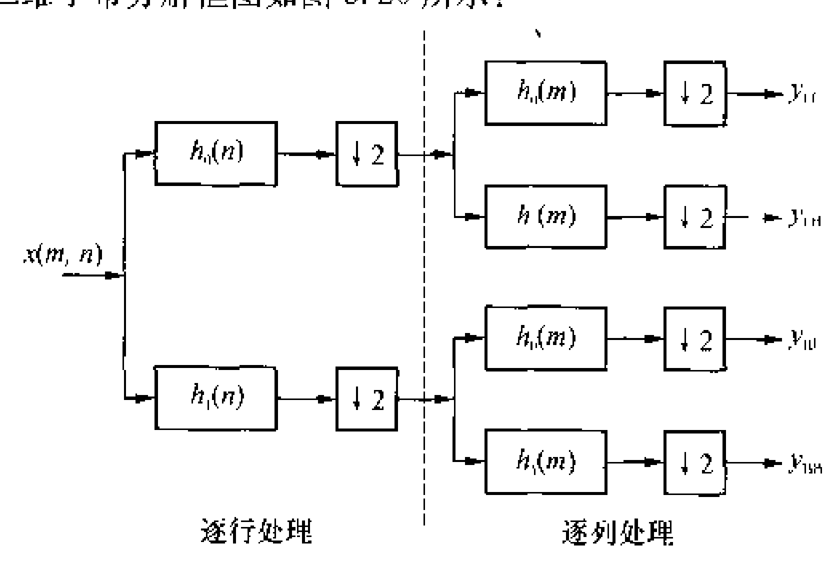


图 3.20　图像的二维子带分解框图

将图像先在行方向上，每行分解为低/高半带信号，然后再在列方向上，每列分解为低/高半带信号，于是一幅图像分解为四个子带图：

$$\begin{cases}\text{LL:} & \text{水平方向低频、垂直方向低频的分量图}\\ \text{LH:} & \text{水平方向低频、垂直方向高频的分量图}\\ \text{HL:} & \text{水平方向高频、垂直方向低频的分量图}\\ \text{HH:} & \text{水平方向高频、垂直方向高频的分量图}\end{cases}$$

在空间域和频率域都可以表示成一分为四的分解过程，实际上在频率域子带间存在(也允许)交叠，某些频率成分在两个子带分量中都存在，并且会有非线性失真项。而且，LL子图还可再一分为四；如此层层分解，分辨率越来越低，形成多级子带，每层的每个分量代表了特定方向，特定分辨率(带通)的信息。

2. 无失真复原条件

上面只是说了可以这样做(分解滤波，亚抽样，补零，滤波合成，结果无冗余，无失真)，对于为什么这样做可以恢复原信号并未说明。

仍回到一维情况来看问题，图3.21是一维信号子带分解和合成过程的频域表示。

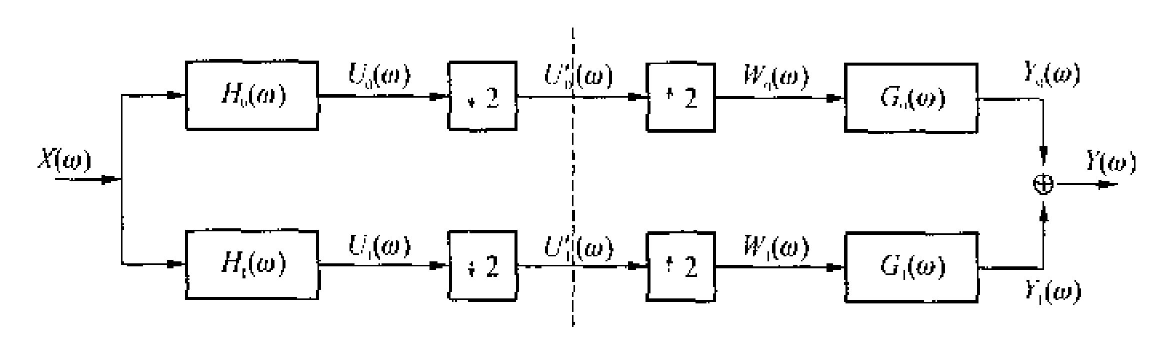


图3.21 一维信号子带分解合成过程的频域表示

若图3.21没有亚抽样和补零，则它成为一个线性系统，只要 $G_0(\omega)H_0(\omega)+G_1(\omega)H_1(\omega)=1$，即可以无失真恢复原信号 $(Y(\omega)=X(\omega))$。

有了亚抽样，特别是对高频分量 $U_1(\omega)$ 亚抽样，明显违背抽样定理，这样做还能不因为混叠而引起失真吗？从图3.22可知，对理想矩形频响半带滤波器可以做到无失真。

但是，实践中不可能有理想滤波器(矩形、半带)，我们总希望滤波器的冲激响应为有限长度。

而且对图像而言，经理想滤波器有Gibbs效应。实际应用中，滤波器 $H_0(\omega)$，$H_1(\omega)$ 近似是矩形半带滤波器，但有“拖尾”。这样，就会因混叠产生新的频率成分，进入(高/低)另一半频带，只靠收端 $G_0(\omega)$，$G_1(\omega)$ 是滤不掉的。这时，我们能否无失真地复原 $X(\omega)$？实际上如果滤波器设计得好仍可无失真复原，关键是让两

路因为混叠造成的新的频率分量正好对消。这时我们无法依靠图例进行说明，需要推导无失真复原条件。

推导：以 $w_k(n)$ 表示经亚抽样，补零后的各路恢复信号，$k=0, 1$，则

$$w_k(n)=\begin{cases}2u_k(n), & n\text{ 为偶数时}\\ 0, & n\text{ 为奇数时}\end{cases}$$

即 $w_k(n)=[1-(-1)^n]u_k(n)$，这就是亚抽样，补零的公式表达。

对上式作 z 变换，用 z 变换的性质，有 $(-1)^n x(n)\leftrightarrow X(-z)$，且由 $U_k(z)=H_k(z)X(z)$，有

$$W_k(z)=U_k(z)+U_k(-z)=H_k(z)X(z)+H_k(-z)X(-z)$$

$$Y_k(z)=G_k(z)W_k(z)=G_k(z)H_k(z)X(z)+G_k(z)H_k(-z)X(-z) \tag{3.7-18}$$

$$\begin{aligned}Y(z)&=Y_0(z)-Y_1(z)\\&=[G_0(z)H_0(z)+G_1(z)H_1(z)]X(z)\\&\quad+[G_0(z)H_0(-z)+G_1(z)H_1(-z)]X(-z)\end{aligned}$$

用 $e^{j\omega}$ 代替 z，以 $e^{j(\omega-\pi)}$ 代替 $-z$，则有

$$\begin{aligned}Y(\omega)=&[G_0(\omega)H_0(\omega)+G_1(\omega)H_1(\omega)]X(\omega)\\&+[G_0(\omega)H_0(\omega+\pi)+G_1(\omega)H_1(\omega+\pi)]X(\omega+\pi)\end{aligned}$$

为使 $Y(\omega)=X(\omega)$，应满足条件：

$$\begin{cases}G_0(\omega)H_0(\omega)+G_1(\omega)H_1(\omega)=1 & (3.7-19a)\\ G_0(\omega)H_0(\omega+\pi)+G_1(\omega)H_1(\omega+\pi)=0 & (3.7-19b)\end{cases}$$

式(3.7-19a)是无线性失真条件，式(3.7-19b)是没有因混迭引起的非线性失真的条件。

由上可知，只要设计 $G_0(\omega)$，$G_1(\omega)$，$H_0(\omega)$，$H_1(\omega)$ 使之满足这两个条件即可无失真复原 $X(\omega)$，由于有四个滤波器，两个约束条件，而不是从四个条件解出四个频率特性来，所以解空间还有一定的“自由度”。

下面举一个和快速小波变换有关的子带滤波器设计实例。

选定 $\begin{cases}H_1(\omega)=e^{j\omega}H_0(-\omega+\pi)\\ G_0(\omega)=H_0(-\omega)\\ G_1(\omega)=H_1(-\omega)\end{cases}$ 对应的时域表示为 $\begin{cases}h_1(n)=(-1)^{1-n}h_0(1-n)\\ h_0(n)=g_0(-n)\\ h_1(n)=g_1(-n)\end{cases}$

有的书上把 $g_0(-n)$ 和 $g_1(-n)$ 记为 $\bar{g}_0(n)$，$\bar{g}_1(n)$，现在 $g_0(n)$ 和 $g_1(n)$ 是实数，所以 $g_0(-n)=\bar{g}_0(n)$，即 g_k 和 h_k 形状相同，只是方向相反。若 $h_k(n)$ 对称，则

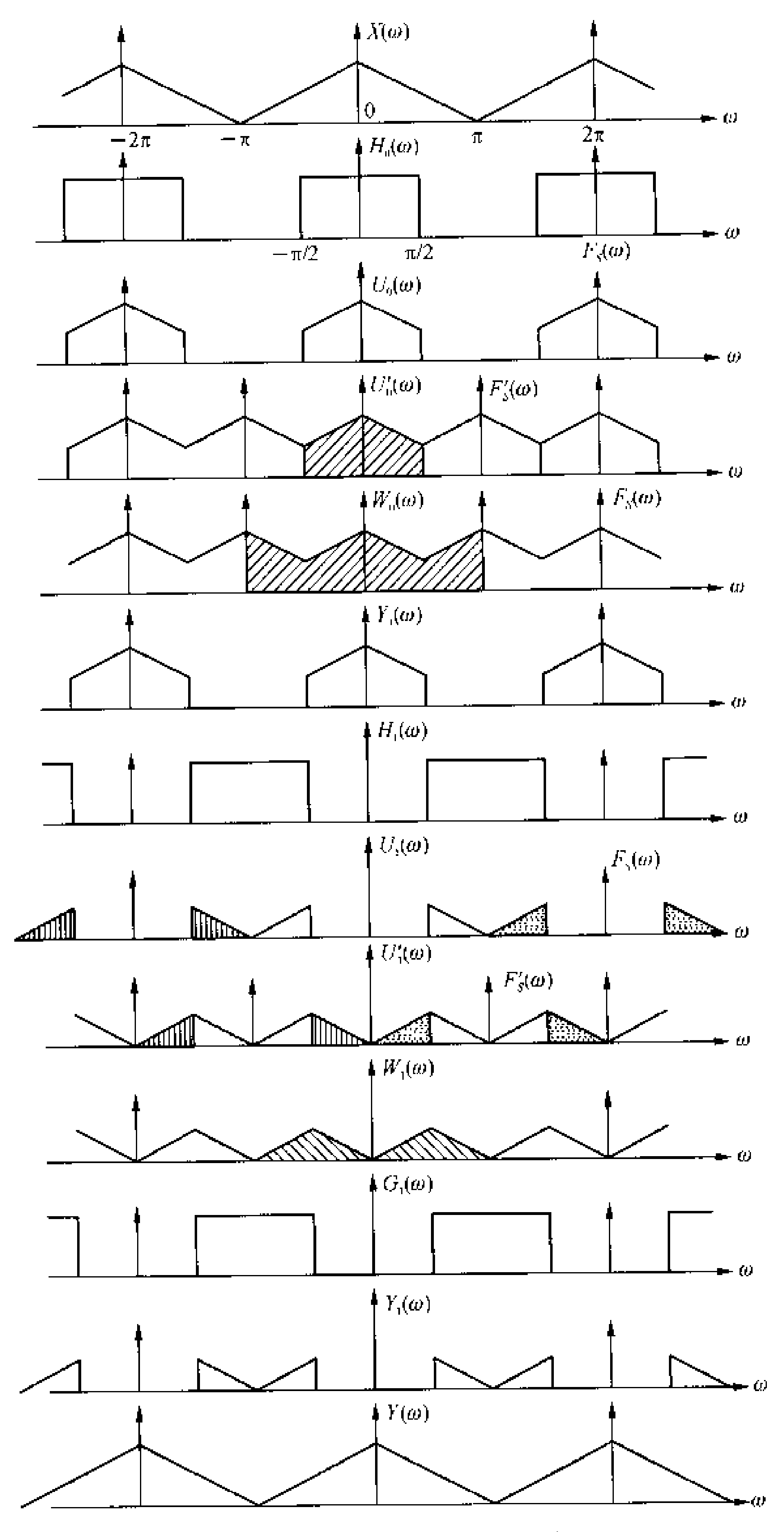


图 3.22　图 3.21 中各点的频谱

$\tilde{g}_k(n) = g_k(n) = h_k(n)$，即收发两端是相同的滤波器。

现在来考察 $H_1(\omega) = e^{j\omega}H_0(-\omega+\pi)$。先不看 $e^{j\omega}$ 这一项，则 $H_0(-\omega+\pi)$ 表示低通变高通，即先翻个身，$H_0(\omega) \to H_0(-\omega)$，再平移半个周期成为 $H_0(-\omega) \to H_0(-\omega+\pi)$。而乘 $e^{j\omega}$ 不会改变 $H_1(\omega)$ 的模，只改变它的相角。

若 $H_1(\omega) = H_0(-\omega)$，则 $h_1(n) = h_0(-n)$；现在 $H_1(\omega) = e^{j\omega}H_0(-\omega+\pi)$，这相当于 $h_1(n) = (-1)^{1-n}h_0(1-n)$，即 $e^{j\omega}$ 相当于在空间域平移一个采样点。

规定了这三个式子之后，相当于把求 G_0，G_1，H_0，H_1 四个子带合成、分解滤波器的问题归结为一个滤波器 $H_0(\omega)$ 的设计问题了。可以发现，不论 $H_0(\omega)$ 怎么取，没有非线性失真的条件总是满足的，即：

$$
\begin{aligned}
& G_0(\omega)H_0(\omega+\pi) + G_1(\omega)H_1(\omega+\pi) \\
&= H_0(-\omega)H_0(\omega+\pi) + H_1(-\omega)H_1(\omega+\pi) \\
&= H_0(-\omega)H_0(\omega+\pi) + e^{-j\omega}H_0(\omega+\pi)e^{j(\omega+\pi)}H_0[-(\omega+\pi)+\pi] \\
&= H_0(-\omega)H_0(\omega+\pi) + e^{j\pi}H_0(\omega+\pi)H_0(-\omega) = 0
\end{aligned}
$$

而要使线性失真为 0（频响要平坦），$H_0(\omega)$ 的设计应满足 $G_0(\omega)H_0(\omega) + G_1(\omega)H_1(\omega) = 1$，即 $H_0(-\omega)H_0(\omega) + H_1(-\omega)H_1(\omega) = H_0(-\omega)H_0(\omega) - H_0(\omega+\pi)H_0(-\omega+\pi) = 1$。低半带分解滤波器冲激响应 $h_0(n)$ 通常取实数，因此 $H_0(\omega)$ 有共轭对称性，即 $H_0(-\omega) = H_0^*(\omega)$。所以上式变形为 $H_0^*(\omega)H_0(\omega) + H_0(\omega+\pi)H_0^*(\omega+\pi) = 1$，或 $|H_0(\omega)|^2 + |H_0(\omega+\pi)|^2 = 1$，此即设计 $H_0(\omega)$ 的要点。把这一类子带滤波器组称为“正交镜像”滤波器组（Quadrature Mirror Filters，QMF）。

进行子带分解时最常用的滤波器就是 QMF。子带分解本身在图像处理中有许多应用，同时小波变换也可用某些特定的子带滤波器组来快速实现。

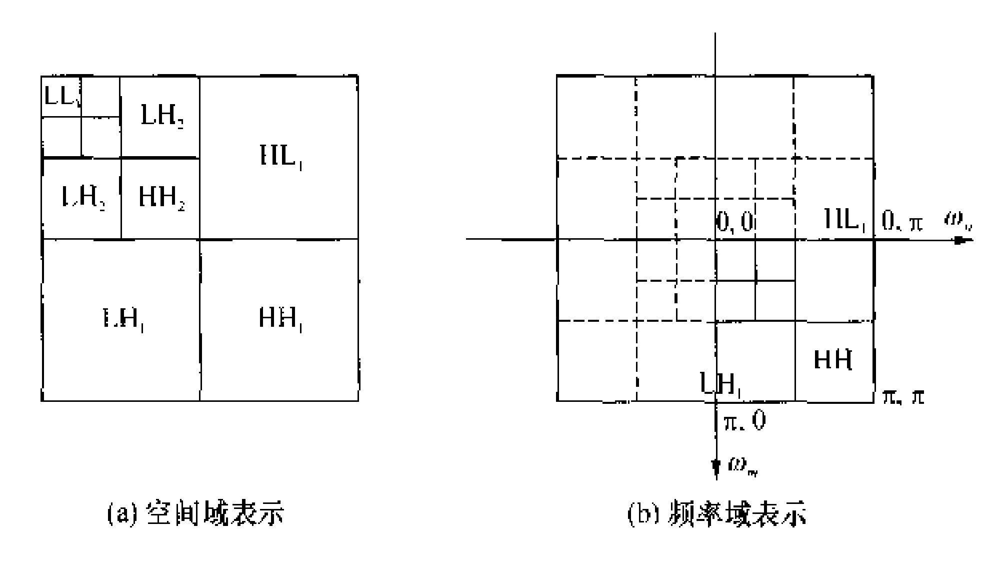
3. 用子带分解实现图像的多分辨率分解

现在来看图像的二维子带分解，对“某一层”的子带分解，可以先逐行再逐列进行高低频半带划分。

而且，可层层分解，即将上一层的低频信号 y_{LL} 再一分为四，得到更低频的子带图。

“空间域”及“频率域”的多层分解如图 3.23 所示，注意频域的“对称性”。

这种方法具有这样一些特点：无失真、无冗余、有方向性、灵活但设计难度高。



(a) 空间域表示 (b) 频率域表示

图 3.23 图像的多层子带分解

3.7.5 多分辨率分析和Mallat的快速小波变换算法

高斯金字塔的启示：它是用多分辨率逐层逼近原始信号的思想来设计的。每层是原始信号(图像是二维信号)在相应分辨率下的"逼近"。上一层可以由下一层经低通滤波和亚抽样获得。

1. 信号 $f(t)$ 在尺度空间 V_j 中的多分辨率逼近[9]

(1) 空间 $L^2(\mathbf{R})$ 的一个多分辨率分析是指满足下列条件的向量空间集 $\{V_j\}_{j\in \mathbf{Z}}$：

① 因果性：$V_j \subset V_{j+1} \quad (j \in \mathbf{Z})$，即 $j+1$ 层处的近似信号包含了所有低于该层分辨率的近似信息(注意这里在层的编号上的表示习惯与金字塔表达上习惯正好相反)。

② 完备性：$\lim\limits_{j\to+\infty} V_j = L^2(\mathbf{R})$，$\lim\limits_{j\to-\infty} V_j = \{0\}$。

③ 尺度不变性：某一近似空间可由其他近似空间经尺度伸缩得到，即 $f(t) \in V_j \Leftrightarrow f(2t) \in V_{j+1} \quad (\forall j \in \mathbf{Z})$。

④ Riesz 基存在性：存在 $\varphi(t) \in L^2(\mathbf{R})$，使函数族 $\{\varphi(t-n) \mid n \in \mathbf{Z}\}$ 是 V_0 的 Riesz 基，即 $\{\varphi(t-n)\}_{n\in \mathbf{Z}}$ 可以张成 V_0 空间，对它规范化可得到 V_0 的规范正交基 $\{\phi(t-n)\}_{n\in \mathbf{Z}}$；$\phi(t)$ 称为尺度函数，它平移成 V_0 的规范正交基。由尺度不变性，从 $\phi(t)$ 的缩放和平移可得各层 V_j 的规范正交基，$\{\phi_j(t-n)\}_{n\in \mathbf{Z}} = \{2^{\frac{j}{2}}\phi(2^j t-n)\}_{n\in \mathbf{Z}}$。

(2) 任一函数 $f(t) \in L^2(\mathbf{R})$ 在多分辨率空间 V_j 中的逼近 $A_j f$ 就是它在 V_j 的基上的正交投影之和(其中 $A_j f$ 是在 2^j 分辨率下逼近函数 $f(t)$ 的算子，是 f 在 V_j 上的正交投影算子，含义相当于"第 j 层高斯金字塔")，即

$$A_j f = \sum_{n\in \mathbf{Z}} a_{j,n}\phi_j(t-n) \tag{3.7-20}$$

$$a_{j,n} = \langle f(t),\ \phi_j(t-n)\rangle = f(t) * 2^{\frac{j}{2}}\phi(-2^j t),\ (t=2^{-j}n) \tag{3.7-21}$$

就是将信号 $f(t)$经滤波(此处为低通)后以 2^{-j} 的采样率进行采样,如 $j=-1$,则是隔点亚抽样;如 $j=-2$,是每四点采一点。这样得到的 $A_j f$ 不再含有分辨率高于 2^j 的信息。记 $a_j=\{a_{j,n}\}_{n\in \mathbf{Z}}$ 称为信号 f 在空间 V_j 中的离散逼近。

(3) 层间递推关系。

由下层逼近信号 $a_{j+1,n}$ 推求上层的逼近信号 $a_{j,n}$:

① 因为 $\phi_{j-1}(t)\in V_{j-1}\subset V_j$,所以 $\phi_{j-1}(t)$可以表示为它在下一层空间 V_j 的基 $\{\phi_j(t-n)\}_{n\in \mathbf{Z}}$ 上的投影之和,即相邻层之间尺度函数满足 $\phi_{j-1}(t)=\sum_{k\in \mathbf{Z}} h(k)\phi_j(t-k)$,式中 $h(k)$可看成层间递推低通滤波器的冲激响应,它和 $\phi_j(t)$,$\phi_{j-1}(t)$间的关系可这样求得:

$$\langle \phi_j(t-k),\ \phi_{j-1}(t-n)\rangle = \langle \phi_j(t-k),\ \sum_{m\in \mathbf{Z}} h(m)\phi_j(t-2n-m)\rangle \tag{3.7-22}$$

上式也称扩张方程或两尺度方程。由 $\phi_j(t)$平移的正交性,仅当 $k=2n+m$,即 $m=k-2n$ 时内积项为1,否则为0,所以上式等于 $h(k-2n)$,从而得到等式

$$h(k-2n) = \langle \phi_j(t-k),\ \phi_{j-1}(t-n)\rangle \tag{3.7-23}$$

② 由 $f(t)$先在 V_{j+1} 空间中投影,得 $A_{j+1}f$ (由各 $a_{j+1,n}$ 来表示),再由层间递推,向上一层的空间 V_j 投影求 $A_j f$,即计算各 $a_{j,n}$的过程如下:

$$\begin{aligned} a_{j,n} &= \langle f(t),\ \phi_j(t-n)\rangle = \langle A_{j+1}f,\ \phi_j(t-n)\rangle \\ &= \langle \sum_{k\in \mathbf{Z}} a_{j+1,k}\phi_{j+1}(t-k),\ \phi_j(t-n)\rangle \\ &= \sum_{k\in \mathbf{Z}} a_{j+1,k}\langle \phi_{j+1}(t-k),\ \phi_j(t-n)\rangle \\ &= \sum_{k\in \mathbf{Z}} h(k-2n)a_{j+1,k} = \sum_{k\in \mathbf{Z}} \bar{h}(2n-k)a_{j+1,k} \end{aligned} \tag{3.7-24}$$

这里 $\bar{h}(n)=h(-n)$ 表示反褶。由(3.7-24)可知,由 a_{j+1} 推求 a_j 的过程就是先用冲激响应为 $\bar{h}(n)$的低通滤波器对逼近信号 a_{j+1} 作滤波卷积,然后再作2∶1的亚抽样(和金字塔/子带时完全一样)。

(4) 若把原数字信号序列 $f(n)$看成在 V_0 空间上的投影(逼近),即令 $f(n)=a_{0,n}$。只要给定低通滤波器 $\bar{h}(n)$,则可以根据上式层层递推地求出 $a_{-1,n}$,$a_{-2,n}$,…,$a_{-J,n}$,即可完成函数 f 在尺度空间中的多分辨率分解过程 $A_0 f\to A_{-1}f\to A_{-2}f\cdots\to A_{-J}f$。

(5) 小结：

上面的叙述说明存在这样的函数——尺度函数 $\phi(t)$，它满足多分辨率分析的"4 性"要求。它的平移形成 V_0 上的规范正交基 $\{\phi(t-n)\}_{n\in Z}$，它缩放后的平移系列则构成其他分辨率下 V_j 空间的规范正交基 $\{\phi_j(t-n)\}_{n\in Z}$，$f(t)$ 在 V_j 空间的逼近就是它在 V_j 上的投影。上面的推导说明了可以不必求出 $\phi(t)$ 和构造各层的正交基 $\{\phi_j(t-n)\}_{n\in Z}$，只要有滤波器冲激响应 $\bar{h}(n)$，就可以通过滤波和亚抽样逐层求得 A_jf，这和子带/金字塔分解时的情况十分相像。$\bar{h}(n)$ 和 $\phi(t)$ 是紧密相关的，它是对同一事物的两种描述方式，但以 $\bar{h}(n)$ 形式表示和处理问题更为方便。

到现在为止，只给出了在尺度空间进行多分辨率分析的数据处理流程的框架，对尺度函数的性质、构造方式并未涉及。事实上，除了要满足多分辨率分析的 4 性外，它们还可以（也应该）满足其他的一些性质。这些要求是相互矛盾的，需要折衷考虑。有许多研究小组对此进行了研究，研究成果以 $\bar{h}(n)$ 的形式列出供选用。

$\phi(t)$ 呈低通特性，它与"高斯金字塔"中的高斯滤波器对应，某尺度下 $\phi_j(t)$ 的滤波输出与"高斯金字塔"中的某层近似信号对应。那么丢失的"细节"到哪里去了？下面讲述的小波空间 W_j 回答了这个问题。

2. 小波空间 W_j 和离散二进正交小波变换的快速实现算法[10]

看相邻两层"逼近"（近似）信号之间的差异：

$V_j \subset V_{j+1}$，故在 V_{j+1} 中存在 V_j 的正交补空间 W_j，使得 $V_{j+1} = V_j \oplus W_j$。其中，V_j 对应于 $f(t)$ 在 2^j 分辨率下的近似；小波空间 W_j 对应于 $f(t)$ 在 2^j 分辨率下的细节信号，因此也称细节空间。从 $j=-\infty$ 起，一层一层地做下去，细节越来越"丰富"，当 $j\to+\infty$ 时，能表示任何 $f(t)\in L^2(\mathbf{R})$ 的能量有限信号。

下面的一系列想法和做法类似于"尺度空间"的多分辨率分析方法，只不过低通特性的尺度函数 $\phi(t)$ 换成了带通特性的 $\psi(t)$，以表示不同分辨率下的"细节"：

滤波器低通 $\bar{h}(n) \to$ 带通 $\bar{g}(n)$

逼近 $A_jf \to$ 细节 D_jf

离散逼近 $a_{j,n} \to$ 离散细节 $c_{j,n}$

分析过程略，与子带分解中的"高半带"概念类比易理解分析结果。可以证明：

存在函数 $\psi(t)$，即母小波。它的平移和缩放构成细节空间 W_j 的规范正交基，即当 $\psi_j(t-n) = 2^{\frac{j}{2}}\psi(2^jt-n)$ 时，$\{\psi_j(t-n)\}_{n\in Z}$ 构成 W_j 的规范正交基。

$f(t)\in L^2(\mathbf{R})$ 在 W_j 上的投影为 $D_jf = \sum_{n\in Z} c_{j,n}\psi_j(t-n)$，其中

$$c_{j,n} = \langle f(t),\ \psi_j(t-n)\rangle \tag{3.7-25}$$

将每层的 D_jf 加起来（j 从 $-\infty$ 变化至 $+\infty$，$j\in\mathbf{Z}$），可表示任何 $f(t)\in L^2(\mathbf{R})$，即

$$f(t)=\sum_{j\in Z}c_{j,n}\psi_j(t-n) \qquad (3.7-26)$$

考虑到W_{j+1}是V_j的子空间，所以W_{j+1}的基$\psi_{j+1}(t)$可以用V_j的基$\phi_j(t)$的加权和来表示，即

$$\psi_{j+1}(t)=\sum_{k\in \mathbf{Z}}g(k)\phi_j(t-k) \qquad (3.7-27)$$

相应地，离散细节

$$c_{j,n}=\sum_{k\in \mathbf{Z}}g(k-2n)a_{j+1,k}=\sum_{k\in \mathbf{Z}}\tilde{g}(2n-k)a_{j+1,k} \quad j,n\in \mathbf{Z} \qquad (3.7-28)$$

即由更细一层的逼近信号a_{j+1}经带通滤波器$\tilde{g}(n)$滤波后再亚抽样，可以获得W_j的离散细节$c_{j,n}$。

现在考虑由逼近信号和细节信号合成原始信号的"反变换"的快速实现：由$V_{j+1}=V_j\oplus W_j$，或$A_{j+1}f=A_jf+D_jf$，即

$$\sum_{k\in \mathbf{Z}}a_{j+1,k}\phi_{j+1}(t-k)=\sum_{k\in \mathbf{Z}}a_{j,k}\phi_j(t-k)+\sum_{k\in \mathbf{Z}}c_{j,k}\psi_j(t-k) \quad (3.7-29)$$

将等式两边对$\phi_{j-1}(t-n)$求内积，考虑到ϕ_{j+1}的正交性，跨层的$\langle\phi_j,\phi_{j-1}\rangle$为低通$h$，$\langle\psi_j,\phi_{j-1}\rangle$为高通$g$，经代数运算得

$$a_{j+1,n}=\sum_{k\in Z}a_{j,k}h(n-2k)+\sum_{k\in \mathbf{Z}}c_{j,k}g(n-2k) \qquad (3.7-30)$$

3. 结果归纳

一维信号的多分辨率分解—合成快速算法（在相邻层之间）可用图3.24表示：

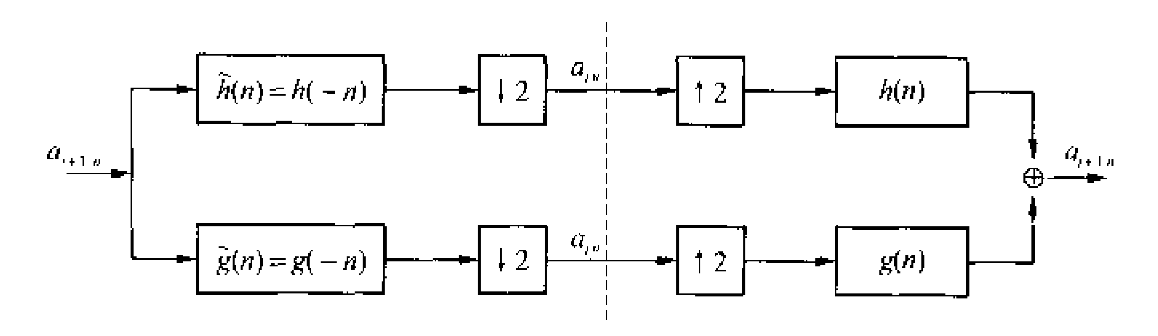


图3.24　一维信号的小波分解合成快速算法框图（Mallat算法）

说明：

(1) Mallat把小波变换、多分辨率分析和子带分解/合成联系起来了（从多分辨率分析和从子带分解两个不同的角度和方法出发，得到相同的结果）。

(2) 图3.24中分解滤波器是合成滤波器的"反向"，因此实际上只需规定两个

滤波器的冲激响应 $h(n)$ 和 $g(n)$ 即可。且因为 g 是 h 的"高半带",可以用 $g(n) = (-1)^{1-n}h(1-n)$ 来从 $h(n)$ 求得 $g(n)$,所以最后只要规定一个滤波器 $h(n)$ 就行了。

由图 3.24 看小波变换,它相当于子带分解/合成的"特例"。

需要认真设计图 3.24 中的滤波器 $h(n)$,这个滤波器除要满足多分辨率分析的 4 性条件,使之能无失真复原外,还要满足其他实际应用中的要求,设计难度较大,实用时常常用"双正交小波变换",放宽了对"正交性"的要求来求得线性相位特性,我们将在下一节对它进行讨论。

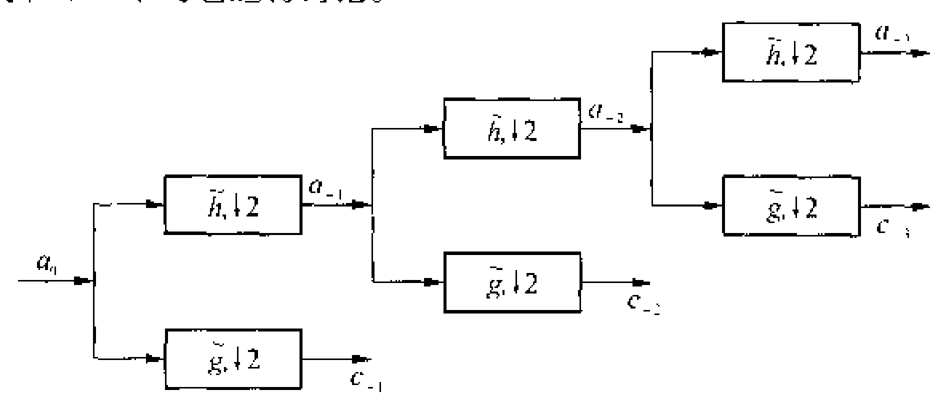


图 3.25 一维信号的多层小波分解

(3) 多次应用上面的框图,将分解后的 a_{j-n} 低频分量(逼近信号)再分解,则构成了称为"鱼骨形"的滤波器组结构。由最上面的路径,即"低通、亚抽样"可得逐级"逼近"信号 a_0, a_{-1}, a_{-2}, a_{-3}, …;由"每节"的分叉可得相应分辨率下的带通信号(细节信号) c_{-1}, c_{-2}, c_{-3}, …,如图 3.25 所示。

(4) 离散二进小波变换是"非冗余"的,数据量没有上升。

4. 图像的快速小波变换(1D→2D)

二维的图像信号,可以先"逐行"作一维小波变换,分解为 L, H 两个分量,再"逐列"作一维小波变换,分解为 LL, LH, HL, HH 四个分量。L 和 H 分别表示低通和高通滤波输出。

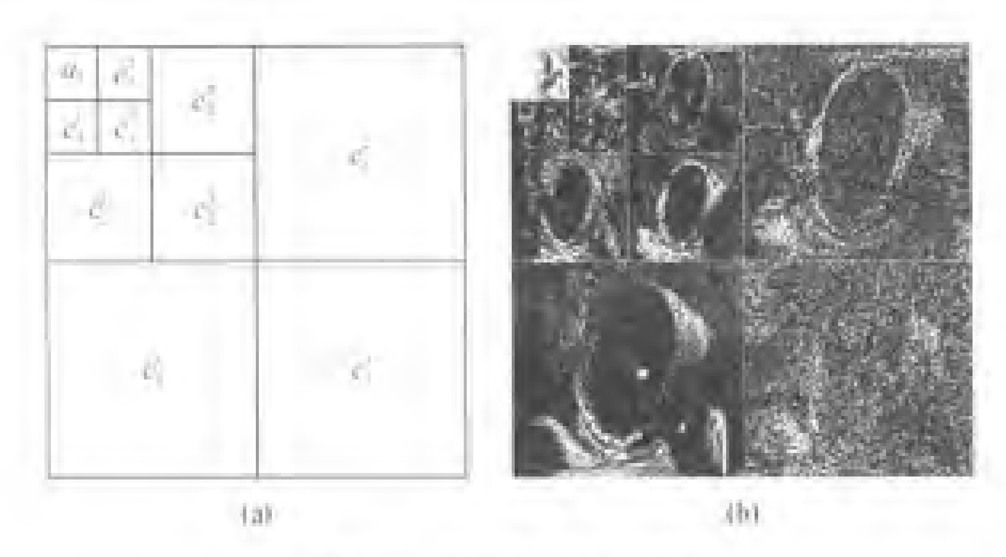
二维尺度函数 $\phi(x, y) = \phi(x)\phi(y)$。

二维小波函数有三个,对应不同方向上的高/低通滤波特性:

$$\begin{cases} \psi^1(x, y) = \phi(x)\psi(y), \ \text{LH} \\ \psi^2(x, y) = \psi(x)\phi(y), \ \text{HL} \\ \psi^3(x, y) = \psi(x)\psi(y), \ \text{HH} \end{cases} \tag{3.7-31}$$

分解的结果,在 2^j 层次有 $A_j f$(逼近)及 $D_j^1 f$, $D_j^2 f$, $D_j^3 f$ 三个细节信号。所以每"上"一层,近似图像分解为四个分量。若原图像为 A_0,分解总层数为 J,则共有 $3J+1$ 幅子图像。

图 3.26 示出 $J=3$(即三次分解)的情况。



(a)　　　　(b)

图 3.26　图像的3层小波分解

共有 $3J+1=10$ 个子图(1 个低通,9 个带通子图)。

从图 3.26(b)中的 a_3 可以看到概貌图,而从其他子图中只能看到边缘和点等细节。

分解合成过程可以表示为:

$$分解:A_{j-1}f \to (A_j f, (D_j^1 f, D_j^2 f, D_j^3 f))$$

$$合成:A_{j-1}f = A_j f + D_j^1 f + D_j^2 f + D_j^3 f$$

图 3.27 给出图像信号的快速小波分解框图。

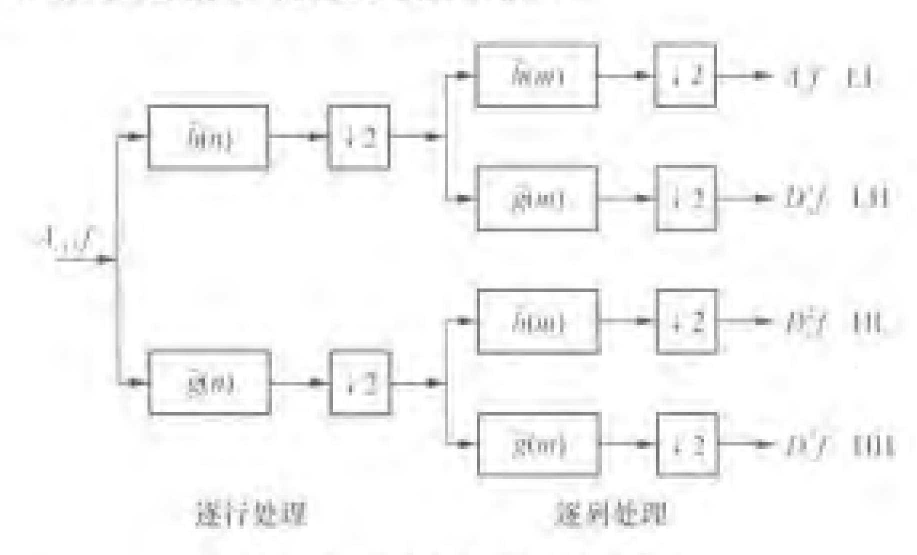


图 3.27　图像的快速小波分解框图

5. 小波基的选择

上面已经对小波变换的信号处理流程进行了详细说明,如今只剩下一个重要问题:怎么选小波基的形状,使它具备好的性能。因为 $\phi(t)$、$\varphi(t)$、$h(t)$、$g(t)$四者

是相互关联的，选定一个另三个也就相应确定了，现在就看母小波 $\psi(t)$ 应该满足什么条件。

以压缩应用为例，小波基除了满足允许条件外，还应当满足：

(1) 有高阶消失矩，即随着 j 增大(分辨率的提高)，小波变换系数 $c_{j,n}$ 迅速减小。这是因为如果高频细节减小迅速，有利于压缩。

(2) 越光滑越好(正则性)。

(3) 两边衰减越快越好(即紧支，支集要小)。

(4) 线性相位特性(最好是零相位)，可使"层间"对准，边缘处没有信号失真，有利于压缩。

上述(1)(2)和(3)要求是矛盾的，小波的消失矩和正则性越好，它的支集越长。而要求(3)和(4)是矛盾的，Daubechies 证明了除 Haar 小波基外，紧支集正交小波必为不对称的，而要相位线性(最好是零相位)，则滤波器的冲激响应应是对称的。

选择时一般最重视紧支性和线性相位性，但在"正交小波基"的限制下，两者很难兼顾。

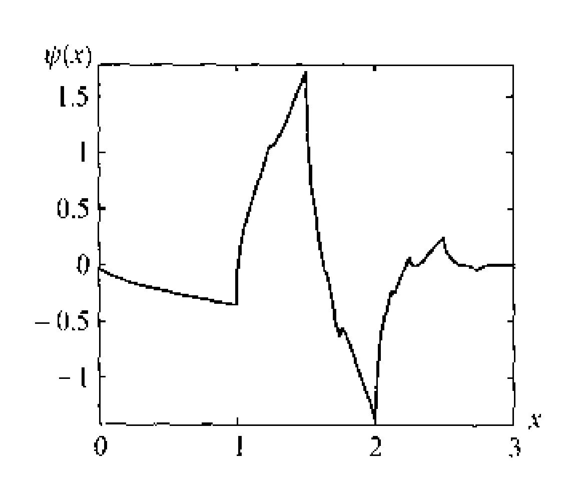


图 3.28 Daubechies 提出的 db2 小波基的波形

Daubechies 提出了一系列不对称、紧支、正交的小波基 db$n$(n 为消失矩)，图 3.28 给出了 db2 小波基的波形图。

用滤波器组的形式来表示的 db2 正交小波基见表 3.5。

表 3.5 db2 正交小波基

n	0	1	2	3	4
$h(n)$	0.1601	0.6938	0.7243	0.1384	−0.2423
n	5	6	7	8	9
$h(n)$	−0.0322	−0.0775	−0.0624	−0.0125	0.0333

3.7.6 双正交小波变换

1. 问题的提出

若分解合成滤波器有非线性相位特性，就会有相位畸变，造成上层图逐层有失真，所以虽能无失真恢复原图，但层间位置关系被破坏，不利于图像的压缩和处理。而且，我们进行图像的小波变换压缩编码时，常常希望层间是零相移，亦即希望

$h(n)$对称，但除 Haar 基外，不存在这样的“紧支撑正交小波基”(Daubechies 证明了这一点)。

于是只好放宽要求。就是放宽对正交性的要求，把它放宽为双正交来做到$h(n)$对称。实际上是把4个子带滤波器归结为设计两个对称滤波器，而不像上小节正交小波变换的快速算法那样，把它们归结为一个低通滤波器的设计。这样，自由度大了，容易做到线性相位或零相位特性。

2. 什么是双正交小波基

它包含两个小波基，一个是$\{\psi_{j,m}(x) \mid j, m \in \mathbf{Z}\}$，另一个是$\{\psi^*_{j,m}(x) \mid j, m \in \mathbf{Z}\}$，它们都构成$L^2(\mathbf{R})$中的“线性无关基”。它们自身的伸缩平移系列并不正交，即$\langle \psi_{j,m}, \psi_{k,n} \rangle \neq \delta(j-k)\delta(m-n)$，但两个小波基的伸缩平移系列相互正交，即$\langle \psi_{j,m}, \psi^*_{k,n} \rangle = \delta(j-k)\delta(m-n)$。我们称$\psi$和$\psi^*$为对偶小波基，它们构成了“双正交小波基”$\{\psi_{j,m}\}$，$\{\psi^*_{k,n}\}$。于是问题就从一个正交的函数族$\{\psi\}$扩展为两个自身不正交，但相互正交的函数族$\{\psi\}$，$\{\psi^*\}$。双正交小波也具有多分辨率分析的特点。

$\{\psi\}$和$\{\psi^*\}$各自仍是“基”，即由它们各自仍可张成$L^2(\mathbf{R})$中不同分辨率的子空间，只不过它们是“线性无关基”，而不是“正交归一基”，所以求变换系数麻烦些。不仅小波函数ψ，ψ^*，而且尺度函数ϕ，ϕ^*也是不正交的。

设一尺度函数$\phi(x) \in L^2(\mathbf{R})$，它的伸缩平移$\{\phi_{j,m} = 2^j\phi(2^jx - m)\}_{j,m\in \mathbf{Z}}$，构成空间$V_j$的基，它本身不正交，但存在一个对偶函数$\phi^*(x)$，它的伸缩平移后的函数族$\{\phi^*_{j,m} = 2^j\phi^*(2^jx - m)\}_{j,m\in \mathbf{Z}}$与$\{\phi_{j,m}(x)\}_{j,m\in \mathbf{Z}}$正交，即

$$\langle \phi_{j,m}(x), \phi^*_{j,n}(x) \rangle = \delta(n-m) \tag{3.7-32}$$

ϕ及ϕ^*对应的两尺度差分方程为：

$$\begin{cases} \phi_j(x) = \sum\limits_{k\in \mathbf{Z}} h(k)\phi_{j+1}(t-k) \\ \phi^*_j(x) = \sum\limits_{k\in \mathbf{Z}} h^*(k)\phi^*_{j-1}(t-k) \end{cases} \tag{3.7-33}$$

令W_j是V_j的补空间，$W_j \subset V_{j-1}$，则相应的小波对偶基$\psi_j(x)$和$\psi^*_j(x)$应分别满足$\psi_j(x) = \sum\limits_{k\in \mathbf{Z}} g(k)\phi_{j+1}(t-k)$和$\psi^*_j(x) = \sum\limits_{k\in \mathbf{Z}} g^*(k)\phi^*_{j-1}(t-k)$，注意$\psi_j$和$\psi^*_j$是双正交的，即$\langle \psi_{j,m}, \psi^*_{j,n} \rangle = \delta(m-n)$。两者的基$\phi_{j-1}$和$\phi^*_{j-1}$相互正交，$\psi_{j,m}$和$\psi^*_{j,n}$相互正交。

3. 双正交小波变换

有了两组不同分辨率、不同位置的小波基$\{\psi_{j,m}\}_{j,m\in \mathbf{Z}}$和$\{\psi^*_{j,m}\}_{j,m\in \mathbf{Z}}$就可以进

行小波变换了。

函数 $f(x)\in L^2(\mathbf{R})$ 可表示成：

$$f(x)=\sum_{j,m\in\mathbf{Z}}c^*_{j,m}\psi_{j,m}(x) \tag{3.7-34}$$

也可表示成：

$$f(x)=\sum_{j,m\in\mathbf{Z}}c_{j,m}\psi^*_{j,m}(x) \tag{3.7-35}$$

(3.7-34)式两边与 $\psi^*_{l,n}(x)$ 作内积运算，可以得到

$$c^*_{j,m}=\langle f(x),\psi^*_{j,m}(x)\rangle \tag{3.7-36}$$

根据第一种表示方法，给出双正交小波的正、反变换式：

$$\begin{cases}f(x)=\sum\limits_{j,m\in\mathbf{Z}}c^*_{j,m}\psi_{j,m}(x)\\ c^*_{j,m}=\langle f(x),\psi^*_{j,m}(x)\rangle\end{cases} \tag{3.7-37}$$

对于第二种表示方法，可以得到：

$$\begin{cases}f(x)=\sum\limits_{j,m\in\mathbf{Z}}c_{j,m}\psi^*_{j,m}(x)\\ c_{j,m}=\langle f(x),\psi_{j,m}(x)\rangle\end{cases} \tag{3.7-38}$$

$$\begin{aligned}f(x)&=\sum_{j,m\in\mathbf{Z}}\langle f(x),\psi^*_{j,m}(x)\rangle\psi_{j,m}(x)\\&=\sum_{j,m\in\mathbf{Z}}\langle f(x),\psi_{j,m}(x)\rangle\psi^*_{j,m}(x)\end{aligned} \tag{3.7-39}$$

4. 双正交小波情况下的多分辨率分析

同样是利用层层逼近，即利用双尺度方程

$$\begin{cases}\phi_j(x)=\sum h(k)\phi_{j-1}(t-k)\\ \psi_j(x)=\sum g(k)\phi_{j-1}(t-k)\end{cases} \tag{3.7-40}$$

$$\begin{cases}\phi^*_j(x)=\sum h^*(k)\phi^*_{j-1}(t-k)\\ \psi^*_j(x)=\sum g^*(k)\phi^*_{j-1}(t-k)\end{cases} \tag{3.7-41}$$

分解：

$$\begin{cases}a_{j,n}=\sum\limits_{k\in\mathbf{Z}}h(k-2n)a_{j-1,k}\\ c_{j,n}=\sum\limits_{k\in\mathbf{Z}}g(k-2n)a_{j+1,k}\end{cases} \tag{3.7-42}$$

合成：
$$a_{j-1,n}=\sum_{k\in Z}h^*(n-2k)a_{j,k}+\sum_{k\in Z}g^*(n-2k)c_{j,k} \tag{3.7-43}$$

若在分解时使用 h 和 g，则在合成时使用 h^* 和 g^* 。若反过来，分解时用 h^* 和 g^* 求出 $a^*_{j,n}$ 和 $c^*_{j,n}$，则合成时用 h 和 g 求出 $a^*_{j-1,n}$。因为它们对应的小波基是“双正交”关系，而不是“正交”的。图 3.29 示出双正交小波变换快速算法的框图。图中为简洁起见，用 A_jf 统称$\{a_{j,n}\}$，用 D_jf 统称$\{c_{j,n}\}$。

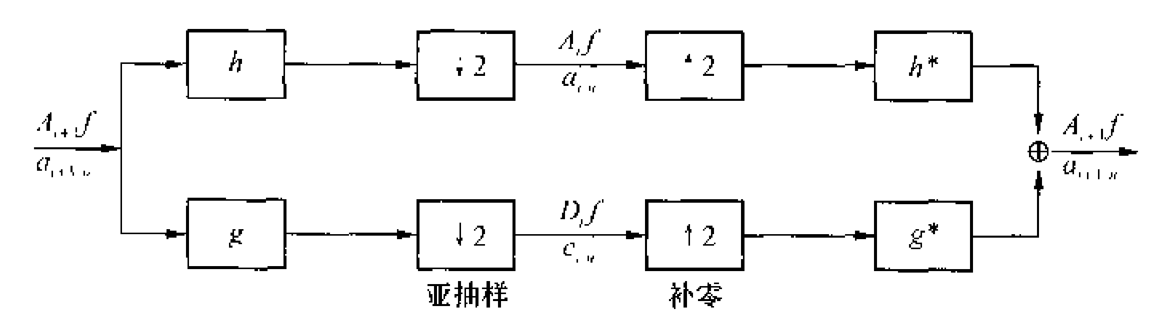


图 3.29　双正交小波变换快速算法的实现框图

说明：

(1) 对照正交小波变换的框图，可见两者是不一样的，收发不是取“反向”，而是另一滤波器。

(2) 对角线上的滤波器成一对，相应的 ϕ 和 ψ^* 双正交，如 h 与 g^* 为“一对”。

(3) 现在可以设计两个对称低通滤波器 h 和 h^*，比“正交”时设计上的自由度大。

(4) 在图 3.29 框图中滤波器的安排也可以是 $h^*\to h$，$g^*\to g$，因为仍然保持“对角”相对应。在选用$\begin{pmatrix}h & h^*\\ g & g^*\end{pmatrix}$形式的安排还是$\begin{pmatrix}h^* & h\\ g^* & g\end{pmatrix}$时可任意。例如在一发多收应用中，为了使接收端简化，可以选取短(抽头少)的滤波器放在收端。

(5) 可以证明，若选 $\begin{cases}g(n)=(-1)^n h^*(1-n)\\ g^*(n)=(-1)^n h(1-n)\end{cases}$ 则可满足无失真条件。

5. 双正交小波基举例：

表 3.6 是 Daubechies 的 9-7 双正交小波变换滤波器组系数表。正因为双正交小波变换可用对称滤波器组成分解-重构滤波器组，较方便地解决了层间失真/

表 3.6　Daubechies 的 9-7 双正交小波变换滤波器组系数表

n	0	±1	±2	±3	±4
$h(n)$	0.8527	0.3774	−0.1106	−0.0238	0.0378
$h^*(n)$	0.7885	0.4181	−0.0407	0.0645	

移位的问题,因此图像的小波变换常常采用双正交小波变换。

3.8 小波包和提升小波

3.8.1 小波包的基本原理及应用

前面讲述的正交小波变换和双正交小波变换的多分辨率分解只将 V(尺度)空间进行了分解,即 $V_0=V_1\oplus W_1=V_2\oplus W_2\oplus W_1=\cdots$,而没有对 W(小波)空间进行进一步分解,各个尺度下时-频相空间的分辨元面积是相等的,即通常所说的等 Q 或倍频程划分。这种等 Q 特性,与人类视觉的分辨特性十分相似,因此正交/双正交小波变换适于视觉处理问题。但在某些场合下,这种时频窗口的固定分布(面积固定)并不是一种最优的选择。在许多问题中,我们只是对某些特定时间段(点)或频域段(点)的信号感兴趣,只需提取这些特定时间及频率点上的信息而已。因此,我们自然希望在感兴趣的频率点上尽可能地提高频域分辨率,而在感兴趣的时间点上则尽可能地提高时间分辨率。

此节我们将要介绍的小波包将对 W_j(小波)空间也作分解,继续将正交/双正交小波变换中随尺度减小而变宽的频谱窗口作进一步的细分,这种优良的性质可允许我们找到最适于待分析信号的时频分辨率,这有关于小波最优基的选取问题。

1. 小波包

可以证明,对任意空间 U 的分解,要求的共轭镜像滤波器关系式与正交小波基时的相同,有关证明可参考文献[11]。在正交小波的二尺度方程中,令正交小波基的滤波器系数分别为 h_n 和 g_n,将尺度函数 $\phi(t)$ 改记为 $u_0(t)$,小波函数 $\psi(t)$ 改记为 $u_1(t)$,给出小波包的定义。

定义:我们称由公式

$$\begin{cases} u_{2n}(t)=\sqrt{2}\sum\limits_{k\in Z}h_k u_n(2t-k) \\ u_{2n+1}(t)=\sqrt{2}\sum\limits_{k\in Z}g_k u_n(2t-k) \end{cases} \tag{3.8-1}$$

定义的函数的集合 $\{u_n(t)\}_{n=0}^{\infty}$ 为由 $u_0=\phi$ 所定义的小波包,它所确定的正交基能改善小波对时间-频域局部化的性能。可以发现,小波包 $\{u_n(t)\}_{n=0}^{\infty}$ 是包括尺度函数 $u_0(\phi)$ 和小波母函数 $u_1(\psi)$ 在内的一个具有一定联系的函数的集合。我们可以

推出 $\{u_n(t)\}_{n=0}^{\infty}$ 小波包函数具有以下性质：

（1）平移正交性：

$$\langle u_n(\cdot-j),\ u_n(\cdot-k)\rangle = \delta_{jk},\ j,\ k\in \mathbf{Z}$$

（2）u_{2n} 和 u_{2n+1} 相互正交：u_{2n} 与 u_{2n+1} 类似于 ϕ 与 ψ 之间的正交关系，满足：

$$\langle u_{2n}(\cdot-j),\ u_{2n+1}(\cdot-k)\rangle = 0,\ j,\ k\in \mathbf{Z},\ n=0,\ 1,\ 2,\ \cdots$$

2. 小波包的基

（1）固定尺度的小波包基。

首先，给出小波库的概念，我们称由标准正交化的多尺度生成元 $\phi(t)$ 导出的函数族 $\{2^{-\frac{j}{2}}u_n(2^{-j}t-k),\ n=1,\ 2,\ \cdots,\ k\in \mathbf{Z}\}$ 为由 $\phi(t)$ 导出的小波库。若定义空间

$$U_j^n = \mathrm{Clos}_{L^2(R)}\{2^{-\frac{j}{2}}u_n(2^{-j}t-k),\ k\in \mathbf{Z}\} \tag{3.8-2}$$

即空间 U_j^n 为由函数 $u_n(t)$ 在 j 尺度下的整数平移系列之线性组合所生成的子空间在 $L^2(R)$ 中的闭包。由于 $u_n(t)$ 是正交平移的，因此 $u_n(t)$ 在 j 尺度下的整数平移系列 $2^{-\frac{j}{2}}u_n(2^{-j}t-k),\ k\in \mathbf{Z}$ 构成空间 U_j^n 的一组正交基。且依此定义，U_0^0 即为零尺度空间 V_0，U_0^1 为尺度为零的小波空间 W_0。

定义：从小波库 $\{2^{-\frac{j}{2}}u_n(2^{-j}t-k),\ n=1,\ 2,\ \cdots,\ k\in \mathbf{Z}\}$ 抽取的能组成 $L^2(R)$ 的一组正交基被称为 $L^2(R)$ 的一个小波包基。选取不同的小波包基，对应于 $L^2(R)$ 的不同分解形式，也即对应于信号不同形式的分解。在小波库中，我们固定尺度 $j=j_0$，可得到 $L^2(R)$ 的一个正交基组 $\{2^{-\frac{j_0}{2}}u_n(2^{-j_0}t-k),\ n=1,\ 2,\ \cdots,\ k\in \mathbf{Z}\}$，并称其为固定尺度的小波包基，与其对应的信号分解形式称为完全分解。假设 $j_0=3$，以二进树的形式给出一维信号的3层分解过程如图3.30所示（二维信号则分解出4个子节点）。

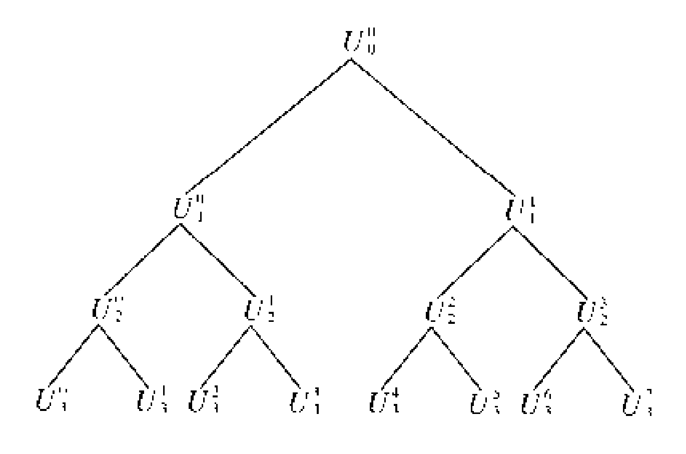


图3.30　一个完全分解的小波包树

当一维信号按照上图的形式作完全分解时，等价于将该信号的频率空间等间隔分割成 2^L 份，其中 L 是小波包的分解层数。对于任一尺度 $0 \leqslant j \leqslant L$，$\{U_j^{p_j}\}_{p_j}^{2^j-1}$，均可用于重构原始信号，它们所对应的小波包基为固定尺度 j。

（2）变尺度的小波包基。

固定尺度的小波包基对信号的频率域作等间隔的完全分解，实质上小波包的分解方式非常灵活，等间隔分解只是其中的一种。现在我们把小波包基推广到混合尺度的情形，为此我们先引入“允许树”的概念。

在一个二进树中，一个节点或可以分解，从而产生两个子节点；或终止分解，成为叶节点，这样的树构成一个“允许树”。任一个“允许树”对应着一种小波包分解，一个子空间 $U_j^{p_j}$，$p_j \in \{0, 1, \cdots, 2^j-1\}$ 在第 j 层不再继续分解，它就构成了允许树的一个叶节点。上图给出的完全分解的小波包树就是一个“允许树”的特例。除了最大尺度外，每一个节点都作了分解。

在一个完整的小波包树中，将所有节点表示的子空间集合 U_0^0，$\{U_1^{p_1}\}_{0\leqslant p_1\leqslant 1}$，…，$\{U_j^{p_j}\}_{0\leqslant p_j\leqslant 2^j-1}$，…，$\{U_L^{p_L}\}_{0\leqslant p_L\leqslant 2^L-1}$ 总称为小波包空间库；相应的 $\{U_j^{p_j}\}_{0\leqslant j\leqslant L,\, 0\leqslant p_j\leqslant 2^j-1}$ 称为小波包基库；各子空间系数集成为小波包系数库。一个分解层深度为 J 的库，可以构造 $2^{2^J-1} \leqslant B_J \leqslant 2^{\frac{5}{4}\cdot 2^J-1}$ 种小波包分解，这就留给我们一个问题：哪种分解是最优的？

（3）最优小波包基。

假设任一函数 $f(t) \in L^2(R)$ 在空间 U_j^n 的投影为 $G_j^n f(t)$，则有

$$G_j^n f(t) = \sum_k d_k^{j,n} \cdot 2^{-\frac{j}{2}} u_n(2^{-j}t - k) \tag{3.8-3}$$

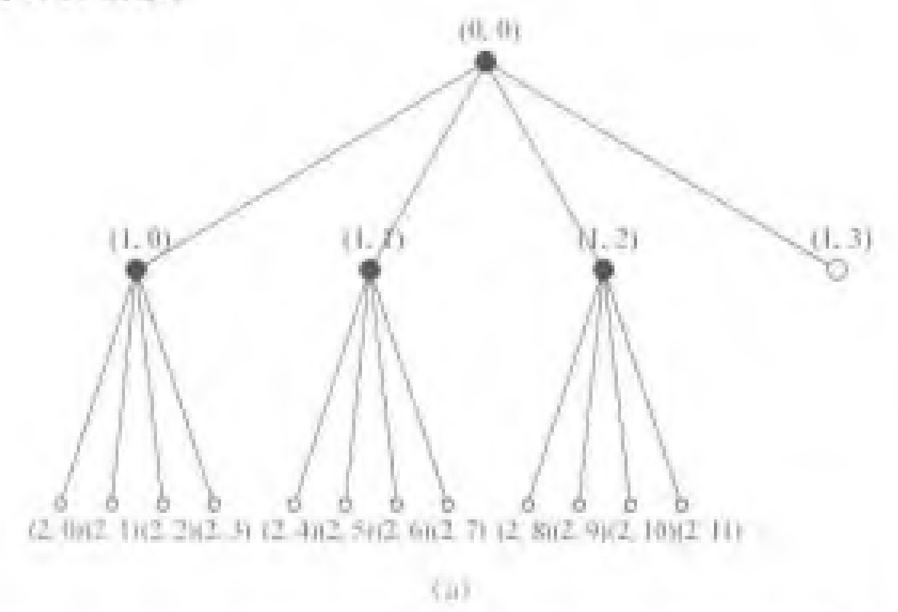
前面提及，小波包对任意空间 U 的分解，要求的共轭镜像滤波器关系式与正交小波基时的相同。回顾正交小波变换的分解过程，并根据二进树的分解中存在 $G_j^n f(t) = G_{j+1}^{2n} f(t) + G_{j+1}^{2n+1} f(t)$，给出小波包系数递推公式：

$$\begin{cases} d_k^{j+1,2n} = \sum_l h(2k-l) d_l^{j,n} \\ d_k^{j+1,2n+1} = \sum_l g(2k-l) d_l^{j,n} \end{cases} \tag{3.8-4}$$

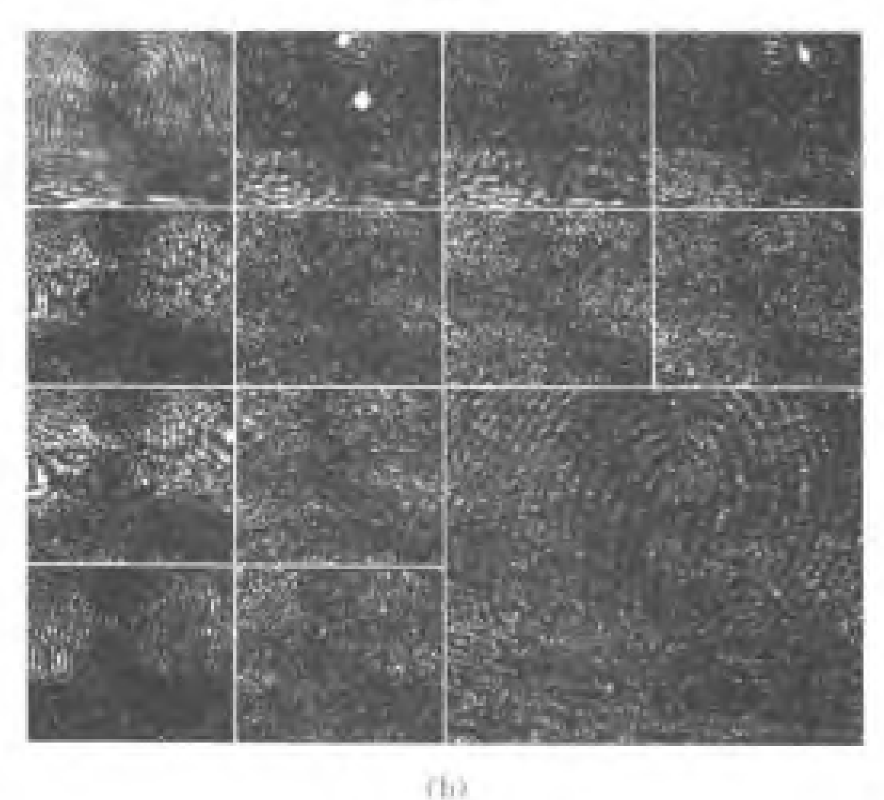
其中，$h(n)$ 和 $g(n)$ 是分解小波包基的冲击响应序列。我们希望小波包分解系数之间的差别越大越好，这样采用少数几个系数就可以描述 $f(t)$ 的特征，显然这样的小波包基是最优的基。

要刻画系数序列 $d_l^{j,n}$ 的性质，首先要定义一个序列的代价函数，然后在小波库的所有小波包基中寻找到使代价函数最小的基。一般，代价函数 M 是能测得集中度的可加性代价函数，如香农熵。这里，集中度是指 M 应是系数差异的增函数；可加性是指 M 满足 $M(0) = 0$ 和 $M(\{x_i\}) = \sum_i M(x_i)$。这种情形下计算出的最优小

波包基,能以最少的系数来描述当前的图像特征,实现有效的图像压缩。最优小波包分解见图 3.31 所示。



(a)



(b)

图 3.31 最优小波包分解(分解级数=2)

3.8.2 提升小波变换

1. 问题的提出

(1) 前面讨论的小波变换是在欧氏空间内通过母小波的伸缩和平移构造小波

基的，不适合非欧氏空间的应用。例如定义在一个特殊曲面（如球面）上的函数，很难通过伸缩、平移描述这样的函数族。

（2）目前所述的小波变换是实数域的变换，即小波变换系数是实数，这对小波变换应用于图像压缩产生了一定的限制。我们非常希望有一种“整数-整数小波变换”，将整数序列映射为低比特的整数小波系数，并且这种映射是可逆的。

（3）能否找到一种小波变换，它的正向变换结构和反变换结构十分一致，便于设计和实现。

（4）伸缩、平移运算在傅里叶变换域中是一般的代数运算，因此小波变换的大多数性质都是通过傅里叶变换工具来描述的。我们能否避免使用傅里叶变换，找到一种描述结构简单，运算量低的小波变换，实现更快速的小波变换算法，这在高速处理、移动手持设备、低功耗设备应用中具有很大的吸引力。

基于以上原因，Sweldens W 等提出了提升小波变换，由于其重要性，提升小波变换又被称作第二代小波变换，而前面在欧氏空间应用的传统小波变换则被称作第一代小波变换。

2. 何谓“提升小波”

首先回顾 Haar 正交小波的例子。在一序列中有相邻数据 a, b，经 Haar 小波变换后其低频信号为 $D=\frac{a+b}{2}$，高频信号则为 $H=b-a$。如果不引入新数据，仅对 a, b 更新，即

$$b:=b-a,\quad a:=a+\frac{b}{2} \tag{3.8-5}$$

这里用操作符：=表示对变量的更新/赋值。我们发现，可在自身位置上完成小波变换，而且还大大简化了计算过程，这在复杂的变换中尤为明显。仔细分析，输入信号被分裂成两个序列 a 和 b，高频分量 b 是前一个值对当前值的预测差，而低频分量随后又由当前值及预测差 b 计算出。这样一个分裂、预测和更新的过程即是第二代小波——提升小波变换的基本思想。

（1）分裂。

假定原始数据集为 $a_0=\{a_{0,i},\ i\in \mathbf{Z}\}$，将其分解成两个集合 a_1, d_1 的方法有多种。在实现中，最常用的是惰性小波变换，即类似于前面 Haar 小波的提升过程，将奇数点集合定义为 a_1，将偶数点集合定义为 d_1。

（2）预测。

该步骤主要是消除第一步分裂后留下的冗余，给出更为紧致的数据表示。预测的目的是用 a_1 预测 d_1，预测误差形成新的 d_1，即

$$d_1 := d_1 - P(a_1) \tag{3.8-6}$$

此处，P 是预测算子。当数据序列 a_1 和 d_1 相关性较大时，通过预测，使更新后的 d_1 具有很低的能量分布。

(3) 更新。

上述分解过程可以继续进行，将 a_1 进一步分解成 a_2 和 d_2，一直将分解过程执行至指定的层数 n，形成系数集合 $\{a_n, d_n, \cdots, d_1\}$。我们希望该过程具有一定的约束，能够保留原始序列 a_0 的某些全局特征，如均值不变，而更新步骤的目的就是使某一全局性质能得以保障。

定义一个标量运算符 $G(a_i)$，由它计算一个数据序列的某个全局量，比如均值。在分解层 i，更新过程须满足

$$G(a_{i+1}) = G(a_i),\ i \in \{0, 1, 2, \cdots, n-1\}$$

为使正反变换结构一致，定义算子 U，使得 $U(d_i)$ 对 a_i 进行更新后，能保持上式成立，更新过程写成

$$a_i := a_i + U(d_i) \tag{3.8-7}$$

综合(1)(2)(3)步骤，我们给出提升小波的正反变换过程以及提升过程的简单框图如下(图 3.32)：

$$\text{正变换:}\begin{cases}\{a_{i+1}, d_{i+1}\} := \text{Split}(a_i) \\ d_{i+1} := d_{i+1} - P(a_{i+1}) \\ a_{i+1} := a_{i+1} + U(d_{i+1})\end{cases} \tag{3.8-8}$$

$$\text{反变换:}\begin{cases}a_{i+1} := a_{i+1} - U(d_{i+1}) \\ d_{i+1} := d_{i+1} + P(a_{i+1}) \\ a_i = \text{Join}(a_{i+1}, d_{i+1})\end{cases} \tag{3.8-9}$$

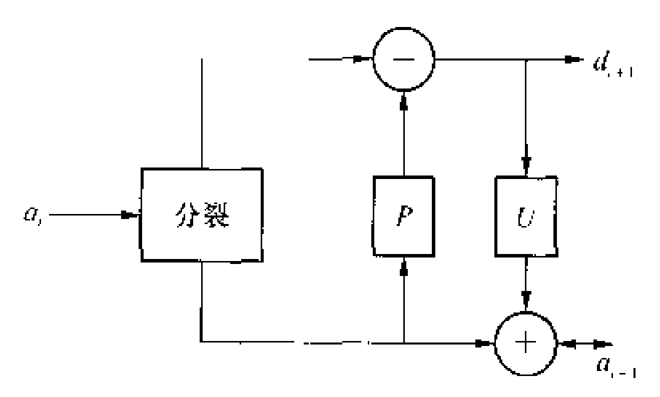


图 3.32 提升过程的简单框图

可以发现，提升方法中的小波正反变换结构仅差一个正负号，且是典型的同址运算，可以实现快速小波变换。提升小波变换过程还摆脱了传统小波的伸缩、平移条件，适于对任一曲面上定义的函数作出小波描述。

3. 提升方法的应用

(1) 构造小波基和实现更快速的小波变换。

在前面讨论双正交小波基的构造时，我们关心的是如何寻找到满足重构条件的一组滤波器$(h, g, \tilde{h}, \tilde{g})$。在利用提升小波构造小波基的问题中，我们关心着能否通过提升方法从已有的滤波器组$(h, g_0, \tilde{h}_0, \tilde{g})$构造出另一组符合给定要求（如具有更高消失矩）的双正交滤波器组$(h, g, \tilde{h}, \tilde{g})$。理论证明，我们可以通过提升方法实现这一点，由于篇幅有限，推导过程不再赘述，下面直接给出提升定理，推导过程可参考文献[12]。

提升定理　取一个初始的双正交尺度函数和小波函数集$(\phi, \psi_0, \tilde{\phi}_0, \tilde{\psi}_0)$，通过提升关系，得到新的函数集$(\phi, \psi, \tilde{\phi}, \tilde{\psi})$，它们是双正交的，提升关系是

空间域：

$$
\begin{aligned}
\psi(x) &= \psi_0(x) - \sum_k s_k \phi(x-k) \\
\tilde{\phi}(x) &= \sqrt{2}\sum_k \tilde{h}_{0,k}\, \tilde{\phi}(2x-k) + \sum_k s_{-k} \tilde{\psi}(x-k) \\
\tilde{\psi}(x) &= \sqrt{2}\sum_k \tilde{g}_{0,k}\, \tilde{\phi}(2x-k)
\end{aligned}
\tag{3.8-10}
$$

频率域：

$$
\begin{aligned}
\hat{\tilde{h}}(\omega) &= \hat{\tilde{h}}_0(\omega) + \hat{\tilde{g}}_0(\omega)\,\overline{\hat{s}(2\omega)} \\
\hat{g}(\omega) &= \hat{g}_0(\omega) - \hat{h}(\omega)\,\hat{s}(2\omega)
\end{aligned}
\tag{3.8-11}
$$

式中，符号^用于描述傅里叶变换域的信号，以区分空间域信号；$\{s_k\}$是可选择的系数；$s(\omega)$是相应的三角多项式，上横线表示共轭。注意，这种提升关系只影响了两个滤波器g和$\tilde{h}$，但影响了三个函数ψ、$\tilde{\phi}$、$\tilde{\psi}$。这是因为虽然$\tilde{g}$没有变化，但是$\tilde{\phi}$的变化导致$\tilde{\psi}$也发生变化。这种ϕ保持不变的提升关系被称为第一类提升，假如是由滤波器组$(h_0, g, \tilde{h}, \tilde{g}_0)$构造出$(h, g, \tilde{h}, \tilde{g})$，即仅对偶尺度函数$\tilde{\phi}$不变，则称其为第二类提升。在构造小波基的实际应用中，我们可以根据提升方程以及给定的构造条件（比如小波基由一阶消失矩变化为二阶消失矩），计算出$s(\omega)$或s_k的解，从而确定出新的小波基。

类似于前面传统小波变换中 Mallat 公式的推导过程，根据提升关系的表达式，可以得到提升方法的快速离散小波变换算法如下：

正变换：

$$
\begin{cases}
a_{i-1,l} := \sum_k h_{k-2l} a_{i,k} \\
d_{i-1,l} := \sum_k g_{0,k-2l} a_{i,k} \\
a_{i-1,l} := a_{i-1,l} + \sum_k s_{l-k} d_{i-1,k}
\end{cases}
\tag{3.8-12}
$$

反变换：

$$
\begin{cases}
a_{i-1,l} := a_{i-1,l} - \sum_k s_{l-k} d_{i-1,k} \\
a_{i,l} := \sum_k h^*_{0,l-2k} a_{i-1,k} + \sum_k g^*_{l-2k} d_{i-1,k}
\end{cases}
\tag{3.8-13}
$$

其中，假定原始滤波器组为(h，g_0，h_0^*，g^*)，新滤波器组为(h，g，h^*，g^*)。由此，提升方法表示的快速离散小波变换框图如图 3.33。

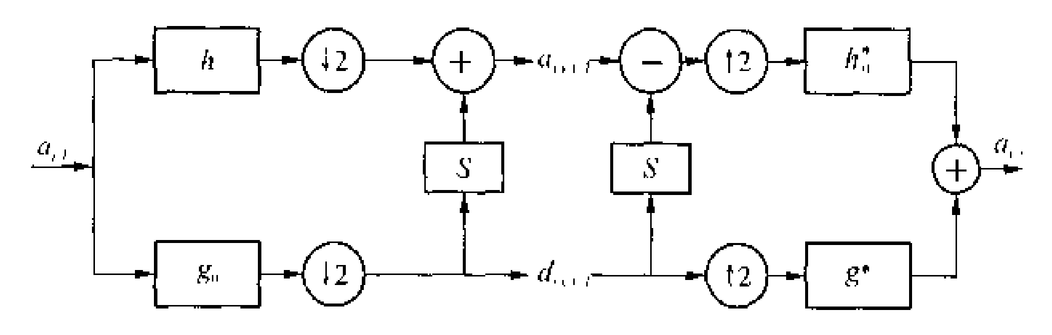


图 3.33 提升方法实现快速离散小波变换

比较双正交小波变换的实现框图，可以看出提升方法中多出了两个提升环节 S。如果新滤波器 g 和 h^* 比原始滤波器 g_0 和 h_0^* 具有更高的消失矩，则提升方法下的总运算量比直接用新的滤波器 g 和 h^* 要低。

(2) 传统小波变换的提升实现。

给定一组双正交小波滤波器(h，g，h^*，g^*)，我们希望得到其传统小波变换的提升实现，以获取更快的变换速度。这就要通过 z 变换和多相表示重新叙述提升过程。首先，给出小波变换的多相实现框图如图 3.34。

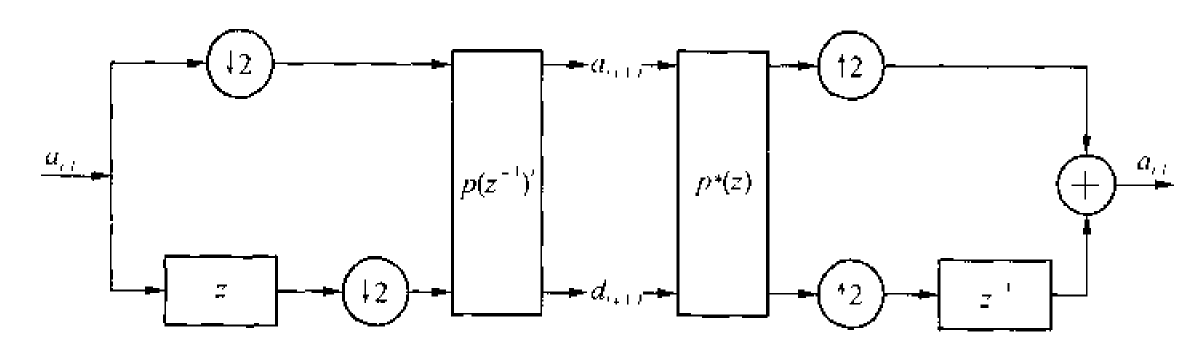


图 3.34 小波变换的多相实现框图

其中，多相矩阵 $[\boldsymbol{p}(z)]=\begin{bmatrix}h_e(z) & g_e(z)\\ h_0(z) & g_0(z)\end{bmatrix}$，$h_e(z)=\sum_k h_{2k}z^{-k}$，$h_o(z)=\sum_l h_{2l+1}z^{-l}$。$[\boldsymbol{p}^*(z)]$ 的定义可类似给出。可以证明，理想重构条件为 $[\boldsymbol{p}^*(z)][\boldsymbol{p}(z^{-1})]^{\mathrm{T}}=\boldsymbol{I}$，$\boldsymbol{I}$ 为单位矩阵。假设提升方法获得的新滤波器组为(h, g_n, h_n^*, g^*)，并以$[\boldsymbol{p}_n(z)]$，$[\boldsymbol{p}_n^*(z)]$表示新的小波变换的多相矩阵，可以证明，当

$$[\boldsymbol{p}_n(z)]=[\boldsymbol{p}(z)]\begin{bmatrix}1 & 0\\ -s(z^{-1}) & 1\end{bmatrix} \tag{3.8-14a}$$

$$[\boldsymbol{p}_n^*(z)]=[\boldsymbol{p}^*(z)]\begin{bmatrix}1 & s(z)\\ 0 & 1\end{bmatrix} \tag{3.8-14b}$$

成立时，(h, g_n, h_n^*, g^*)满足重构条件。其中，$s(z)$为提升系数 s_k 的 z 变换。若用 Euclidean 算法作因式分解，可以得到 $\begin{bmatrix}h_e(z)\\ h_o(z)\end{bmatrix}=\prod_{i=1}^{n}\begin{bmatrix}\boldsymbol{q}_i(z) & 1\\ 1 & 0\end{bmatrix}\begin{bmatrix}k\\ 0\end{bmatrix}$，矩阵$[\boldsymbol{q}_i(z)]$和常数 k 的具体定义可参考文献[13]。那么，$[\boldsymbol{p}(z)]$和$[\boldsymbol{p}^*(z)]$的因式分解结果如下：

$$[\boldsymbol{p}(z)]=\prod_{i=1}^{m}\begin{bmatrix}1 & [\boldsymbol{q}_{2i-1}(z)]\\ 0 & 1\end{bmatrix}\begin{bmatrix}1 & 0\\ [\boldsymbol{q}_{2i}] & 1\end{bmatrix}\begin{bmatrix}1 & k^2s(z)\\ 0 & 1\end{bmatrix}\begin{bmatrix}1 & 0\\ 0 & 1\end{bmatrix}\begin{bmatrix}k & 0\\ 0 & 1/k\end{bmatrix} \tag{3.8-15a}$$

$$[\boldsymbol{p}^*(z)]=\prod_{i=1}^{m}\begin{bmatrix}1 & 0\\ -[\boldsymbol{q}_{2i-1}(z^{-1})] & 1\end{bmatrix}\begin{bmatrix}1 & -[\boldsymbol{q}_{2i}(z^{-1})]\\ 0 & 1\end{bmatrix}\begin{bmatrix}1 & 0\\ -k^2s(z^{-1}) & 1\end{bmatrix}\begin{bmatrix}1 & 0\\ 0 & 1\end{bmatrix}\begin{bmatrix}1/k & 0\\ 0 & k\end{bmatrix} \tag{3.8-15b}$$

式中：$1\leqslant i\leqslant m$，$m=\frac{n}{2}$。观察图 3.34，在通过多相矩阵之前，原始序列被亚采样分裂成两个序列，而多相矩阵按因式分解结果可看成一个级联变换过程，在每一级变换中，序列得以预测或更新，一个较复杂的传统小波变换问题就转化为一个简单的级联提升过程了。

正变换（如图 3.35）：

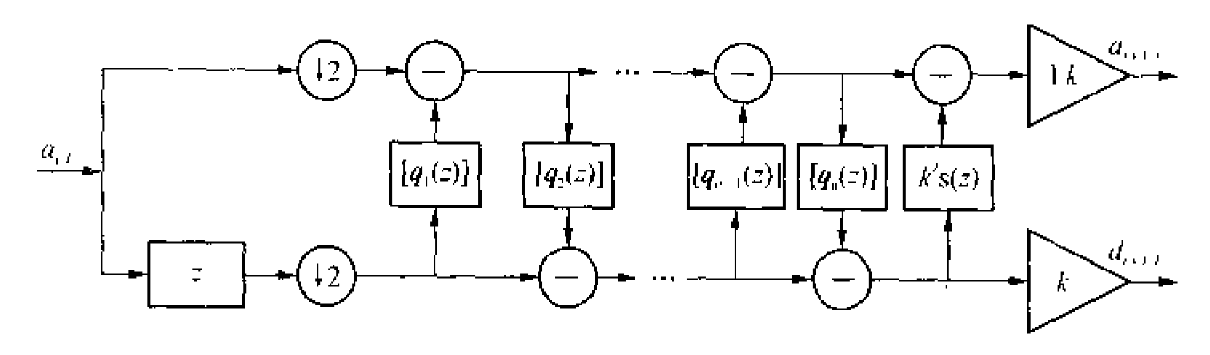


图 3.35 正变换

反变换(如图 3.36):

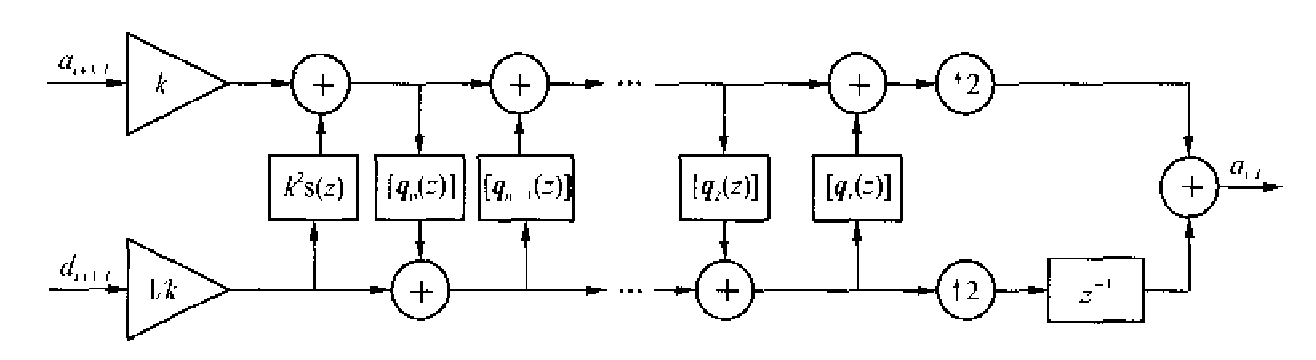


图 3.36　反变换

(3) 构造整数小波变换。

以上分析表明,传统小波变换可以用一个简单的级联提升过程予以描述,因此我们将提升过程稍作变形,给出一级分解的整数小波变换过程:

正变换:

$$\{a_{1,l}^{(0)},\ d_{1,l}^{(0)}\} := \mathrm{split}(a_0) \tag{3.8-16a}$$

$$d_{1,l}^{(i)} := d_{1,l}^{(i-1)} - \left\lfloor \sum_k p_k^{(i)} a_{1,l-k}^{(i-1)} + \frac{1}{2} \right\rfloor \tag{3.8-16b}$$

$$a_{1,l}^{(i)} := a_{1,l}^{(i-1)} + \left\lfloor \sum_k u_k^{(i)} d_{1,l-k}^{(i)} + \frac{1}{2} \right\rfloor \tag{3.8-16c}$$

反变换:

$$a_{1,l}^{(i-1)} := a_{1,l}^{(i)} - \left\lfloor \sum_k u_k^{(i)} d_{1,l-k}^{(i)} + \frac{1}{2} \right\rfloor \tag{3.8-17a}$$

$$d_{1,l}^{(i-1)} := d_{1,l}^{(i)} + \left\lfloor \sum_k p_k^{(i)} a_{1,l-k}^{(i-1)} + \frac{1}{2} \right\rfloor \tag{3.8-17b}$$

$$a_0 = \mathrm{Join}(a_{1,l}^{(0)},\ d_{1,l}^{(0)}) \tag{3.8-17c}$$

可以证明,上述变换是可逆的。这种整数-整数的小波变换无须浮点存储,十分适用于图像压缩编码。

习　　题

1. 求证下列直积的一些性质:

(1) $([\boldsymbol{A}]\otimes[\boldsymbol{B}])([\boldsymbol{C}]\otimes[\boldsymbol{D}]) = ([\boldsymbol{A}][\boldsymbol{C}])\otimes([\boldsymbol{B}][\boldsymbol{D}])$

(2) $T_r([\boldsymbol{A}]\otimes[\boldsymbol{B}]) = T_r[\boldsymbol{A}]\cdot T_r[\boldsymbol{B}]$

(3) $([\boldsymbol{A}]+[\boldsymbol{B}])\otimes[\boldsymbol{C}] = [\boldsymbol{A}]\otimes[\boldsymbol{C}]+[\boldsymbol{B}]\otimes[\boldsymbol{C}]$

(4) $([\boldsymbol{A}]\otimes[\boldsymbol{B}])^{-1} = [\boldsymbol{A}]^{-1}\otimes[\boldsymbol{B}]^{-1}$

以上[A]，[B]，[C]，[D]都是 $N\times N$ 阶矩阵。

2. 已知 $f(n)$，$n=0, 1, \cdots, N-1$。现在把它以 $n=0$ 为对称中心扩充为 $2N-1$ 点，并进行 $2N-1$ 点的傅里叶变换以形成奇数余弦变换。求这个变换在一维中的表达式，以及对应的酉矩阵。并写出可分离核情况下二维变换的表达式。

3. 已知数字图像 $f(m, n)$的二维傅里叶变换 $F(s, t)$的能量集中在频率域的左上，左下，右上，右下四个角附近。问：可以对图像进行怎样的空间域处理，使处理后的图像 $g(m, n)$的二维傅里叶变换 $G(s, t)$的能量集中在 $\left(\frac{M}{2}, \frac{N}{2}\right)$附近。

4. 从概念上说明离散余弦变换比离散傅里叶变换有更强的能量集中现象，因而更适于作图像压缩。

5. 用离散卡-洛变换将四维向量降为二维，若原四维向量的协方差矩阵的特征值为 20，15，10，5，问：要降为二维应舍弃与哪两个特征值对应的特征向量？这样造成的失真是多少？

6. 对题图 3.1 所示的 4×4 图像作二维哈尔变换和二维哈达玛变换，并通过反变换验证其正确性。

1	1	1	1
1	4	4	1
1	3	2	1
1	1	1	1

题图 3.1

7. 对 256×256 大小的灰度图像，用 C++编程实现对其进行 3 层小波分解，并基于小波系数重构图像的程序，分解和重构时采用双正交 9-7 小波。

8. 学习用 Matlab 中的小波库，用其中的函数分别对图像进行小波变换和小波包变换。

参考文献

[1] Jain A K. A sinusoidal family of unitary transform [J]. IEEE Trans. on PAMI-1(4) Oct. 1979, 356~365.

[2] Makhoul John. A fast cosine transform in one and two dimensions [J]. IEEE Trans. on ASSP, Vol. 28, No. 1, Jan. 1980, 27~34.

[3] 阿罕麦德 N，罗著 K R. 数字信号处理中的正交变换 [M]. 胡正名，陆传贵译. 北京：人民邮电出版社，1979.

[4] Jain A K. A fast Kahunen--Loeve transform for a class of stochastic process [J]. IEEE Trans. on Commun. Vol 24, sept. 1976, 1023~1029.

[5] Bracewell R M. The Fourier Transfor an its applications(2nd ed.) [M]. New York: McGraw Hill, 1986.

[6] Pratt W K, Andrews H C, Kane J. Hadamard Transform image coding, Proc [J]. IEEE. Vol. 57, No. 1, 1969, 58~68.

[7] Burt P J, Adelson E H. The laplacian pyramid as a compact image code [J]. IEEE Trans. on Commu. 1983, Vol. 31, No. 4, 532~540.

[8] Antonini M, Barlaud M, et al. Image Coding using wavelet transform [J]. IEEE Trans. on Image Processing. Vol. 1, No. 2, 1992, 205~220.

[9] Akansu A N, Haddad R A. Multiresolution signal Decomposition [M]. Transforms, Subbands, Wavelets, Academic Press, inc, 1992.

[10] Mallat S.. A theory for multiresolution signal decomposition: the wavelet representation [J]. IEEE Trans. on PAMI, 1989, Vol. 11, No. 7, 674~693.

[11] Daubechies I. Orthonomal based of compactly supported wavelets, Commu [J]. in pure and application math., Vol. 41, 1988, 906~996.

[12] Daubechies I, Sweldens W, Factoring wavelet transforms into lifting steps [J]. J. Fourier Anal. Appl. 1998, Vol. 4, 247~269.

[13] 张旭东，卢国栋，冯健. 图像编码基础和小波压缩技术 — 原理、算法和标准[M]. 北京：清华大学出版社，2004.

[14] Kenneth. R. Castleman. 数字图像处理[M]. 朱志刚，等译. 北京：电子工业出版社，1998.

第 4 章　图像增强

图像增强(Enhancement)的目的是为了改善图像的视觉效果，或者是为了更便于人或计算机对图像的分析和处理。在图像增强的过程中突出了一部分信息，同时可能会压制另一部分信息。图像增强与图像复原不同，它在不了解图像降质的具体原因的情况下，用试探的方式对图像进行加工，力求改善图像质量。

图像增强的处理，有的针对图像像素的灰度值，有的针对目标的几何形状，有的以灰度值变化的快慢(即空间频率)作为增强的依据，有的以减少图像上的噪声干扰为主要目标。

目前在图像增强方面还没有统一的质量评价标准。在图像处理理论方面的工作面临着严重的障碍，这种障碍就是缺乏从图像外观的角度进行主观判别的数学度量。这一障碍直接与图像增强的质量评价标准有关。例如，已有一幅图像 $f(m, n)$，要找出两幅图像 $f_1(m, n)$，$f_2(m, n)$中的哪一幅更接近于 $f(m, n)$，目前则没有一种通用的度量方法与人的主观观察结果是一致的。

本章介绍图像增强的若干方法，并进行分析。值得注意的是，图像增强具有方法简单、效果明显的特点，许多方法已经得到了广泛的应用，在数字图像处理中具有十分重要的地位。

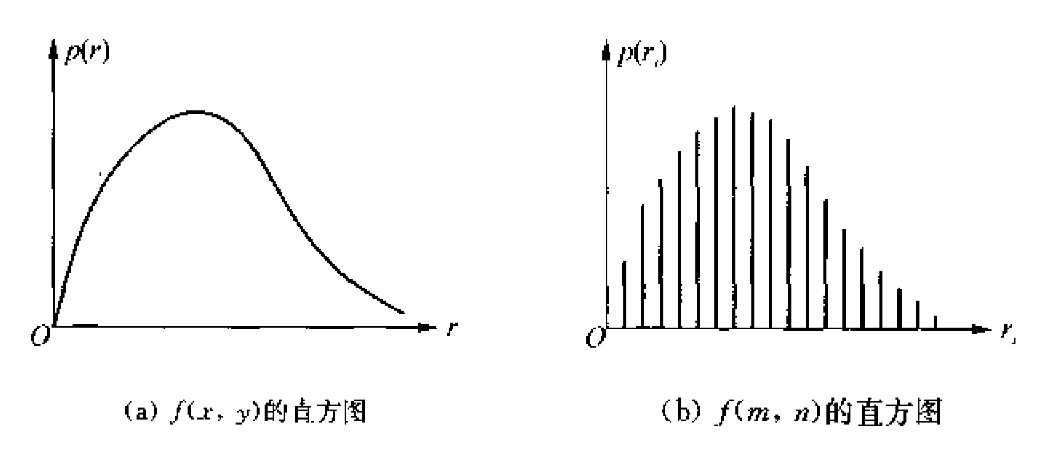
图像增强的方法大致可分为三大类。第一类为点变换，即处理时只与本身像素点的值有关，而与空间邻点的值无关。第二类为空间域处理，在处理中往往作邻域加权和处理，邻域范围称为处理窗。第三类为频域处理，即将 $f(m, n)$变换到频域上，对 $F(s, t)$进行处理，处理后再将 $F'(s, t)$反变换为 $f'(m, n)$进行显示。此外，还有小波去噪、同态滤波、中值滤波、伪彩色、假彩色等处理方法。

4.1　灰度增强

4.1.1　灰度直方图

面积为 A 的连续图像 $f(x, y)$经数字化后，成为 M 行 N 列的数字图像

$f(m, n)$。就数字图像 $f(m, n)$来说，在图像中取不同灰度值的像素的数目是不同的。如果取灰度值大的像素的数量大，则整幅图像看上去就比较亮；反之，则偏暗。直方图是用于表达图像灰度分布情况的统计图表。其横坐标是灰度值 r，纵坐标是出现这个灰度值的概率密度 $p(r)$（对连续图像 $f(x, y)$而言），或者是出现这个灰度值的概率值 $p(r_i)$（对数字图像 $f(m, n)$而言），参见图 4.1。直方图反映了取不同灰度值 r 的面积或像素数在整幅图像上所占的比例，还可以求出一幅图像中所含的信息量；但是直方图排除了取 r 灰度值的像素在图像中位置方面的任何信息，因此直方图并不回答原图中有关目标形状方面的任何问题。



(a) $f(x, y)$的直方图　　(b) $f(m, n)$的直方图

图 4.1　灰度直方图

我们先来看连续图像 $f(x, y)$的直方图。设图像的面积为 A，$A(r)$是灰度值小于 r 的这部分图像的面积和，则概率密度 $p(r)$为：

$$p(r) = \lim_{\Delta r \to 0} \frac{A(r + \Delta r) - A(r)}{\Delta r \cdot A} \tag{4.1-1}$$

且有

$$\int_{r_{min}}^{r_{max}} p(r)\mathrm{d}r = 1 \tag{4.1-2}$$

作出 $r - p(r)$曲线，如图 4.1(a)所示，得到 $f(x, y)$的直方图。

在数字图像 $f(m, n)$的情况下，上述表达式略有不同。设图像像素的灰度值为 $r_0, r_1, \cdots, r_{k-1}$，则概率 $p(r_i)$为：

$$p(r_i) = \frac{\text{灰度值为 } r_i \text{ 的像素数}}{\text{图像上总的像素数}} (i = 0, 1, \cdots, k-1) \tag{4.1-3}$$

且有

$$\sum_{i=0}^{k-1} p(r_i) = 1 \tag{4.1-4}$$

在计算机中可以方便地从 $f(m, n)$求出 $p(r_i)$的分布。由于此时 r_i 的取值是离散的，因此作出的 $r_i - p(r_i)$曲线是不连续的，习惯上画成图 4.1(b)的形式。

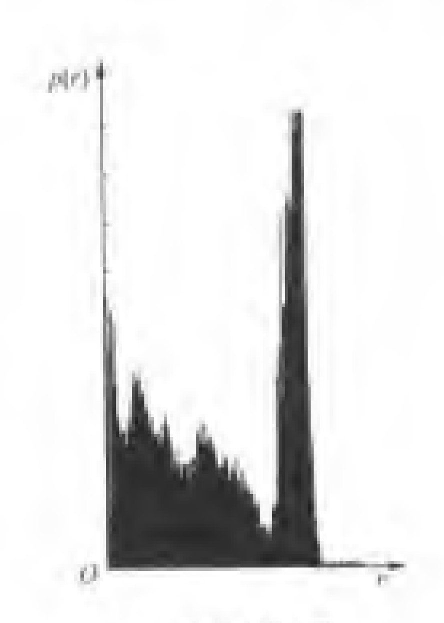
图 4.2 给出了一幅数字图像及其对应的直方图。

灰度增强就是按一定变换关系逐点地改变原来图像中每一个像素的灰度值。设原来图像像素的灰度值为 $r = f(x, y)$，处理后图像像素的灰度值为 $r' = g(x, y)$，灰度增强处理可以表达为：

$$g(x, y) = T(f(x, y)) \quad 或 \quad r' = T(r) \tag{4.1-5}$$



(a) 原始图像



(b) 对应的直方图

图 4.2 数字图像及其对应的直方图举例

确定了灰度变换的关系 $r' = T(r)$，也就确定了一个具体的灰度增强的方法。$r' = T(r)$ 可以看成一个单值函数。

由于像素灰度的取值有了变化，直方图的形状也有了相应的变化。4.1.3 小节中将介绍从直方图形状的改变入手来确定变换关系 $r' = T(r)$ 的方法。

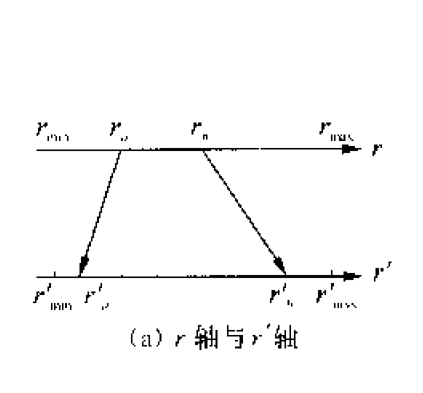
下面以连续图像为例介绍的方法，也适用于数字图像的场合。

4.1.2 灰度的线性变换

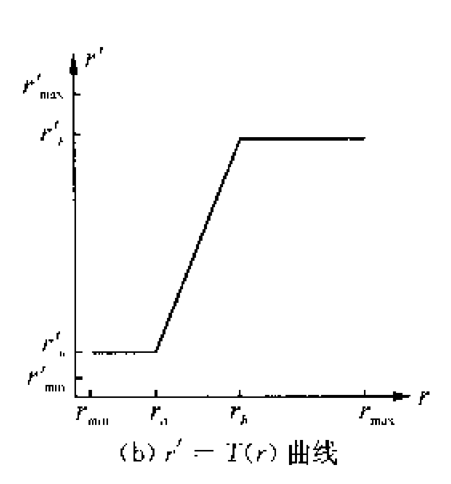
如果 $r' = T(r)$ 是一个线性的或分段线性的单值函数，则由它所确定的灰度变换称为灰度的线性变换。

1. 线性变换

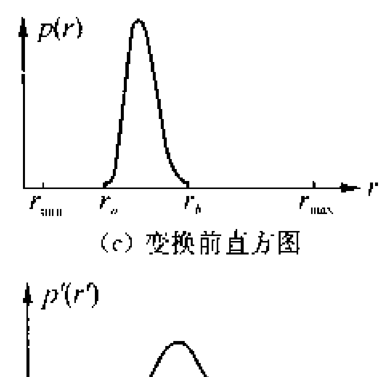
设图像的灰度值 $r = f(x, y)$ 的可能取值区间为$[r_{min}, r_{max}]$。但是由于某种原因，使得取值范围大为缩小，成为$[r_a, r_b]$，其中 $r_{min} \leqslant r_a < r_b \leqslant r_{max}$，如图4.3(a)所示。例如连续图像 $f(x, y)$的值的动态范围过小，或者黑白照片曝光不足，都会造成这种情况。这时对应的直方图 $p(r)$如图 4.3(c)所示，它集中在一个较小的灰



(a) r 轴与 r' 轴



(b) $r' = T(r)$ 曲线



(c) 变换前直方图

(d) 变换后直方图

图 4.3 灰度的线性变换

度区间内。

现在采用这样一种线性变换：

$$
\begin{cases}
r' = \dfrac{r'_b - r'_a}{r_b - r_a}(r - r_a) + r'_a, & r \in [r_a, r_b] \\
r' = r'_a, & r \in [r_{min}, r_a] \\
r' = r'_b, & r \in [r_b, r_{max}]
\end{cases}
\tag{4.1-6}
$$

则可以把变换后的灰度 r' 的取值范围扩大到 $[r'_a, r'_b]$。在极端的情况下 $r'_a = r'_{min}$，$r'_b = r'_{max}$。由式(4.1-6)所确定的变换关系 $r \sim r'$ 曲线如图 4.3(b)所示，进行线性变换之后，灰度 r' 的直方图如(d)所示。

从 2.5.3 小节可知，当两个相邻像素的灰度值(对应着它们的亮度)相差到一定程度时，人眼才会感觉到它们之间亮度上的差别。灰度值 r 仅在较小的区间 $[r_a, r_b]$ 内时，人眼对该图可以感觉到的亮度差的总级数是很少的。这时如果某一目标的灰度值与其背景的灰度值相差得较小时，人眼便无法检测出来。在作了 $r' = T(r)$ 的转换后，如果像图 4.3 那样使 $[r'_a, r'_b]$ 变得更大了，那么人眼对变换后的图像可以感觉到的亮度差的总级数将增加。变换后的图像看上去要清晰得多。

应当说明的是在图像数字化之前还是在数字化之后进行灰度增强，有时结果是不同的。如果对数字化后的图像 $f(m, n)$ 作增强处理，由于像素灰度值 r_i 在区间 $[r_a, r_b]$ 内仅取了 k 个不同的值，经变换后的灰度值 r'_i 也只是在放大了的区间 $[r'_a, r'_b]$ 内至多取 k 个不同的值，只是最大和最小灰度值之差更大了一些。如果在连续图像 $f(x, y)$ 中待检测目标与背景的灰度值之差很小，各自的量化值进入同一灰度级内，则在 $f(m, n)$ 中该目标实际上已经消失，用灰度增强的方法是无法突出该目标的。然而在数字化之前完成对图像 $f(x, y)$ 的增强(由模拟放大器实现)，使得

$$
| r'_b - r'_a | = C \quad r_b - r_a |
$$

情况就会不同。当 $C>1$ 时就有放大作用，把目标和背景的微小差别放大。假设 $C=2$，再数字化，则 $f(m, n)$ 可在 $[r_a', r_b']$ 范围内取值，共有 $2k$ 个不同的灰度值。同一个目标与其背景的灰度值之差就会是 1 级。只要有差别，就可用灰度增强手段把目标突出出来。而在前一种情况下，微弱的目标已在量化过程中丢失了。可见数字图像 $f(m, n)$ 并不完全反映原始的连续图像 $f(x, y)$。在图像进入计算机时，应十分注意从数量对比上保持感兴趣目标的原有信息。如果发生了目标信息丢失的情况，其后的处理可能是无意义的。

另一种线性变换如图 4.4(a)所示，$r \sim r'$ 曲线是负斜率的。其表达式为：

$$r' = r'_{max} - \frac{r'_{max} - r'_{min}}{r_{max} - r_{min}}(r - r_{min}) \tag{4.1-7}$$

它的作用是完成灰度的倒置，即 $f(x, y)$ 中灰度的最大(最小)值转换为 $g(x, y)$ 中的最小(最大)值。在 $f(x, y)$ 中某个像素的灰度值越大，对应在 $g(x, y)$ 中该像素的灰度值越小。它使正片转换成负片(如照相底片)。

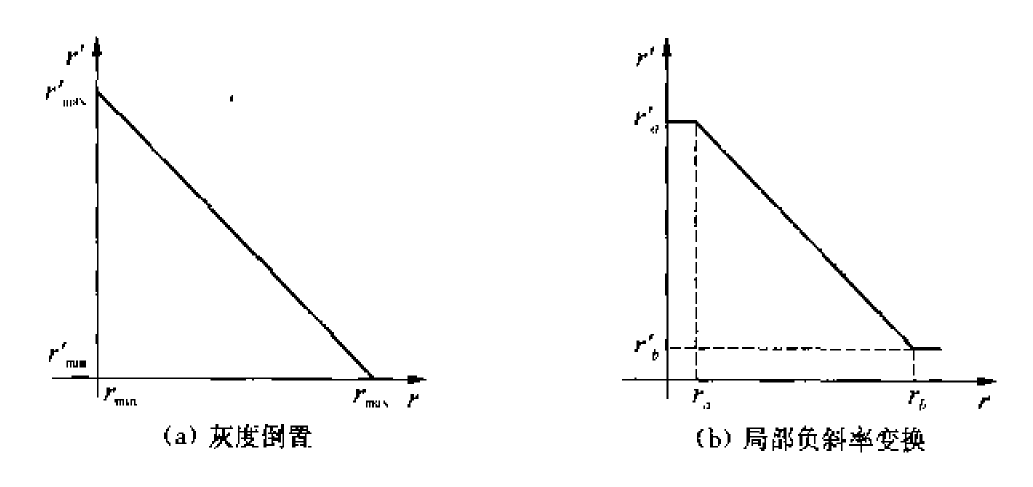


图 4.4　负斜率灰度线性变换

2. 分段线性变换

该图像把 $[r_a, r_b]$ 分成两段 $[r_a, r_c]$、$[r_c, r_b]$ 进行线性变换，就是分段线性变换，有时也将 $[r_a, r_b]$ 分成多段进行线性变换。采用分段线性变换可以压缩一部分灰度区间，扩展另一部分灰度区间。其变换表达式为：

$$\begin{cases} r' = \dfrac{r'_c - r'_a}{r_c - r_a}(r - r_a) + r'_a, & r \in [r_a, r_c] \\ r' = \dfrac{r'_b - r'_c}{r_b - r_c}(r - r_c) + r'_c, & r \in (r_c, r_b] \end{cases} \tag{4.1-8}$$

变换曲线如图 4.5(b)和(d)所示。图中(a)和(b)是扩展第一区间 $[r_a, r_c]$ 而压缩第二区间 $[r_c, r_b]$；(c)和(d)是压缩第一区间 $[r_a, r_c]$ 而扩展第二区间 $[r_c, r_b]$。用多

段折线构成一个单值函数，可以逼近一条曲线，完成较为精细的灰度变换。

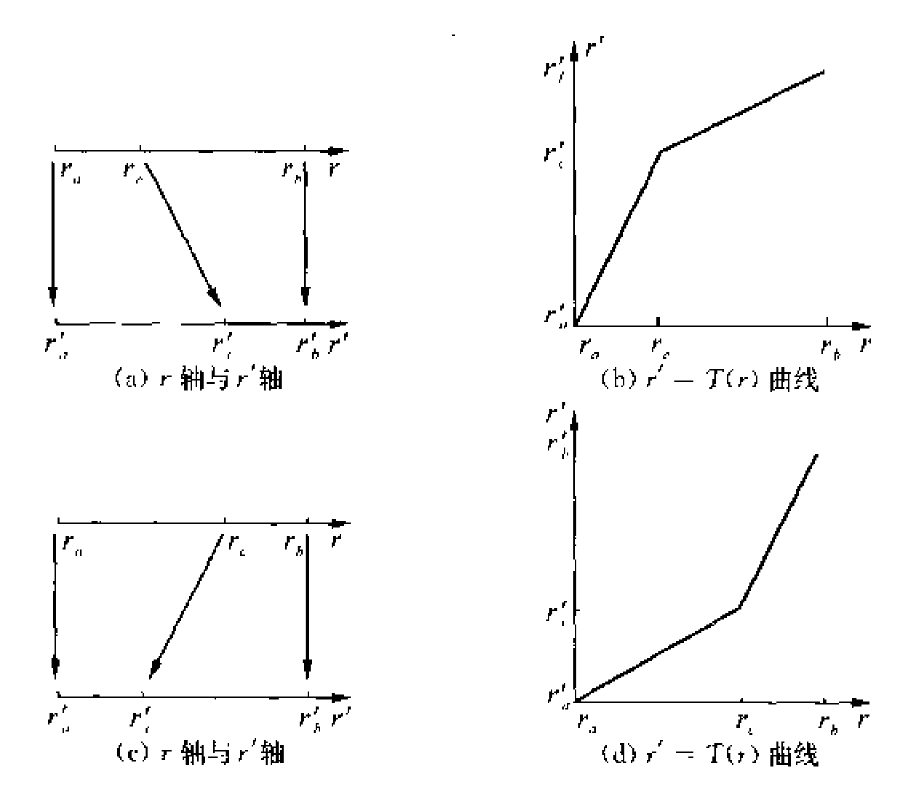


图 4.5　分段线性变换

第二种分段线性变换如图 4.6 所示，又称锯齿形变换，其表达式为：

$$
\begin{cases}
r' = \dfrac{r'_b - r'_a}{r_c - r_a}(r - r_a) + r'_a, & r \in [r_a,\ r_c] \\[2ex]
r' = \dfrac{r'_b - r'_c}{r_b - r_c}(r - r_c) + r'_a, & r \in (r_c,\ r_b]
\end{cases}
\tag{4.1-9}
$$

在显示设备的亮度动态范围不足以满足要求时，可以采用这种变换再显示，以利于在较大的灰度值范围内显示出灰度值的差异。这时 $r' = T(r)$ 是单值的，但是 $r = T^{-1}(r')$ 却是多值的。此外，当锯齿数过多时，将不易从图像上判断原图 $f(m,\ n)$ 中像素原来灰度值的大小。

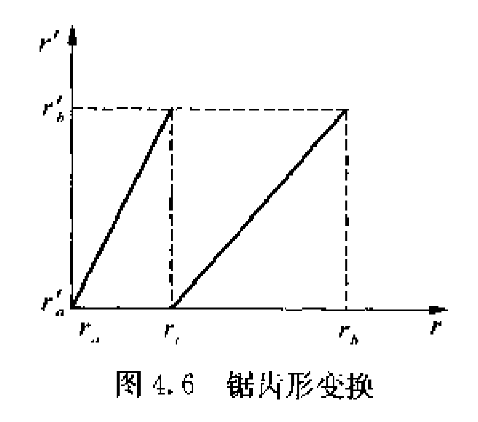


图 4.6　锯齿形变换

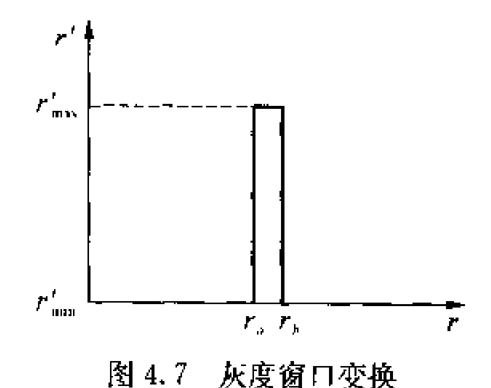


图 4.7　灰度窗口变换

第三种分段线性变换如图 4.7 所示，又称灰度窗口变换。其表达式为：

$$\begin{cases} r' = r'_{max}, & r \in [r_a, r_b] \\ r' = r'_{min}, & r \notin [r_a, r_b] \end{cases} \tag{4.1-10}$$

在实际使用中，子区间$[r_a, r_b]$可大可小，并可位于$[r_{min}, r_{max}]$中的任何位置，故称$[r_a, r_b]$是一个灰度窗口。用这种变换可以检测图像中具有某一个灰度值（当$r_a = r_b$时）的所有像素或某一窗口内所有像素的灰度值，是图像灰度分析的得力工具。

本小节中介绍的几种灰度线性变换的方法可以单独使用，也可以组合起来使用，以满足各种处理中的具体要求。在大多数情况下使用这些方法都可以获得良好的效果。

4.1.3 灰度的非线性变换

如果$r' = T(r)$是一个非线性的单值函数，则由它所确定的灰度变换称为灰度的非线性变换。

在 4.1.1 小节中曾提到，当完成了像素的灰度变换后，图像直方图的形状也有了相应的变化。从图 4.3 中可以看出灰度变换前后的直方图$p(r)$和$p'(r')$形状是不同的。因此，也可以从直方图变化（或称直方图修正）的角度来讨论灰度变换。下面先讨论连续图像的灰度非线性变换，再把结论推广到数字图像的情况。

1. 连续图像灰度的非线性变换

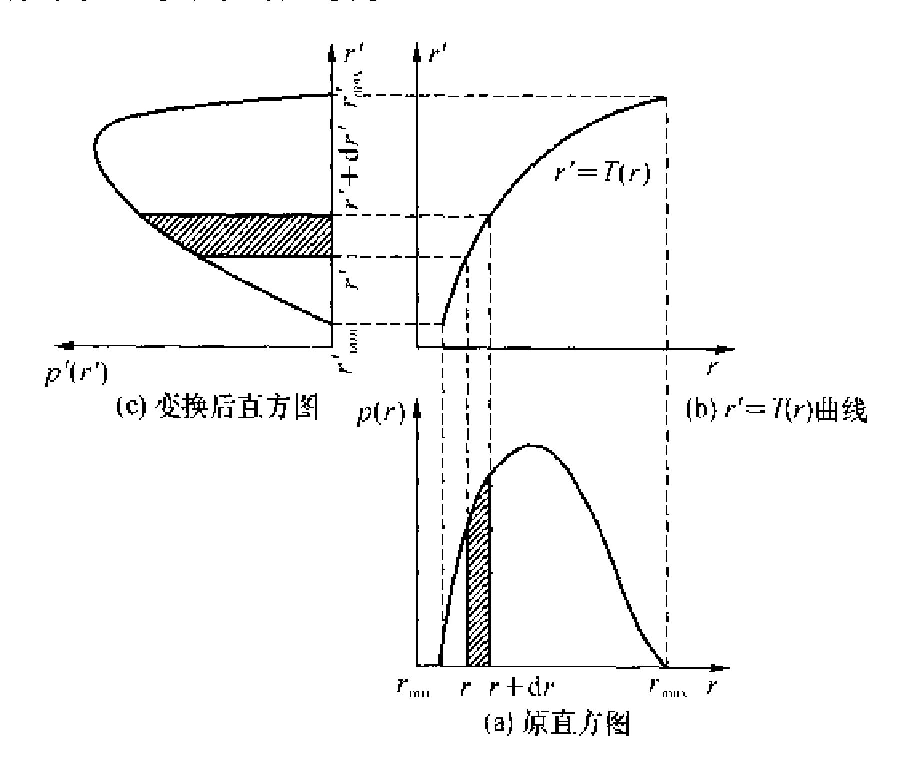


图 4.8　$f(x, y)$图像灰度的非线性变换

设有一连续图像,在坐标(x, y)处的灰度值为$r = f(x, y)$。在$r \in [r_{min}, r_{max}]$时,r的直方图为$p(r)$。经过由$r' = T(r)$确定的灰度变换后得到连续图像$g(x, y)$。对应有在$r' \in [r'_{min}, r'_{max}]$时,$r'$的直方图为$p'(r')$。各符号的含义见图4.8。直方图应满足(4.1-2)式,因此有:

$$\int_{r_{min}}^{r_{max}} p(r)\mathrm{d}r = 1 \tag{4.1-11}$$

$$\int_{r'_{min}}^{r'_{max}} p'(r')\mathrm{d}r' = 1 \tag{4.1-12}$$

在下面的讨论中,下述条件是已知的:变换前灰度r的取值范围$[r_{min}, r_{max}]$、变换后灰度r'的取值范围$[r'_{min}, r'_{max}]$、原图$f(x, y)$及其直方图$p(r)$以及指定的$g(x, y)$的直方图$p'(r')$。现在要求的是灰度的变换关系$r' = T(r)$。而$g(x, y)$则要灰度变换完成之后才可得到。$r' = T(r)$应当是一个单调、连续、可微函数。为了不至于处理后得到一个负像(灰度倒置),它还应当是一个单调增函数。这样反函数$r = T^{-1}(r')$也是一个单调增函数。

图像$f(x, y)$中,有一定比例的面积,其灰度值落在区间$[r, r+\mathrm{d}r]$中。根据直方图的含义,图像中这部分面积所占整个图像面积的比例为$p(r)\mathrm{d}r$。根据式(4.1-11),灰度值落在全部$[r_{min}, r_{max}]$范围中的面积的比例为

$$\int_{r_{min}}^{r_{max}} p(r)\mathrm{d}r = 1$$

即整个图像面积的比例值为1。

灰度变换之后,$f(x, y)$中的灰度范围$[r, r+\mathrm{d}r]$对应着$g(x, y)$中的灰度范围$[r', r'+\mathrm{d}r']$,见图4.8。类似地,在图像$g(x, y)$中对应有一定比例的面积,其灰度值落在区间$[r', r'+dr']$中,这部分面积所占比例为$p'(r')\mathrm{d}r'$。

因为直方图仅反映取不同灰度值的面积或像素在整幅图像上所占的比例,所以,对直方图进行灰度变换时不会改变对应的面积或像素的比例。即不论$T(r)$取何值,灰度变换都只是把原图中灰度值在$r \sim r+\mathrm{d}r$之间的这部分图的灰度值变为$r' \sim r'+\mathrm{d}r'$之间。由于这种变换是灰度值一对一的变换,不改变具有这种灰度值的面积的大小,因此有:

$$p(r)\mathrm{d}r = p'(r')\mathrm{d}r' \tag{4.1-13}$$

式中:$p(r)$和$p'(r')$是已知的,该式以微分形式隐含了r与r'的对应关系。

图像$f(x, y)$中具有$[r_{min}, r]$范围内那些灰度值的图像面积,占有整幅图像面积的比例为:

$$F(r)=\int_{r_{\min}}^{r} p(r)\mathrm{d}r,\ r_{\min}\leqslant r\leqslant r_{\max} \tag{4.1-14}$$

它对应着 $g(x, y)$ 中具有 $[r'_{\min}, r']$ 范围内那些灰度值的图像面积的比例：

$$F'(r')=\int_{r'_{\min}}^{r'} p'(r')\mathrm{d}r',\quad r'_{\min}\leqslant r'\leqslant r'_{\max} \tag{4.1-15}$$

变换前后，这个面积比例值保持不变，故有 $F(r)=F'(r')$，即：

$$\int_{r_{\min}}^{r} p(r)\mathrm{d}r=\int_{r'_{\min}}^{r'} p'(r')\mathrm{d}r' \tag{4.1-16}$$

这里以积分的形式隐含了 r 与 r' 的对应关系。从式(4.1-16)可以求出变换关系式 $r'=T(r)$，并可由(4.1-13)式验证运算结果。现举例说明求 $r'=T(r)$ 的过程：

设某图的直方图为 $p(r)=-2r+2,\ r\in[0, 1]$。现在要进行一灰度的非线性变换，使变换后的直方图为 $p'(r')=1=$ 常数，$r'\in[0, 1]$，求变换式 $r'=T(r)$。

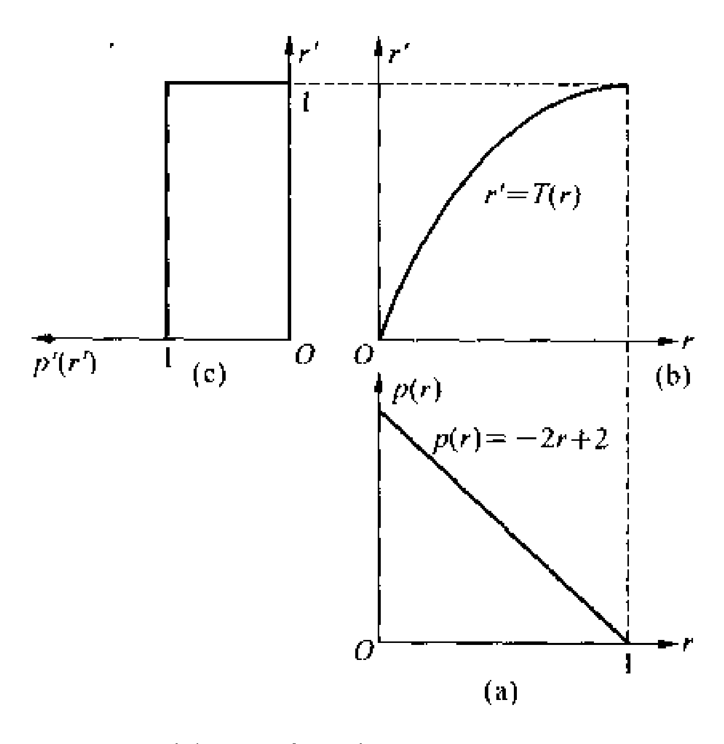


图 4.9 灰度非线性变换例子

由(4.1-16)式可以得到：

$$\int_{0}^{r'} 1\cdot \mathrm{d}r'=\int_{0}^{r}(-2r+2)\mathrm{d}r,\ r'=-r^2+2r,\ r,\ r'\in[0, 1] \tag{4.1-17}$$

由(4.1-13)式可以验证(4.1-17)的结论。图 4.9 中示出了灰度变换前后图像的直方图 $p(r)$ 和 $p'(r')$，图 4.9(b)中的曲线就是灰度变换曲线，它是一个单调增函数。

2. 数字图像灰度的非线性变换

上面得出的结论经过一点修正，可以推广到数字图像的情况。这是一种“点变换”的增强处理方法，即逐点进行灰度变换，不牵涉到邻域像素值的计算。

设有一数字图像在坐标(m, n)处的像素为第k级灰度，即$r_k = f(m, n)$。该图共有K级灰度，即$k = 0, 1, \cdots, K-1$，其直方图为$p(r_k)(k=0, 1, \cdots, K-1)$，简写为$p(k)$。经灰度变换后得到图像$g(m, n)$，在坐标$(m, n)$处像素的灰度为$r'_l$。$g(m, n)$共有$K'$级灰度，即$l = 0, 1, \cdots, K'-1$，其直方图为$p'(r'_l)$，简写为$p'(l)(l = 0, 1, \cdots, K'-1)$。灰度变换中灰度级之间的对应关系写成$l = T(k)$。变换前后两图的灰度级数$K$和$K'$可以相同，也可以不同。

对直方图的约束条件为：

$$\sum_{i=0}^{K-1} p(i) = 1 \tag{4.1-18}$$

$$\sum_{j=0}^{K'-1} p'(j) = 1 \tag{4.1-19}$$

在下面的讨论中，下述条件已知：变换前后的灰度级数K和K'，$f(m, n)$的直方图$p(k)$、指定希望得到的$g(m, n)$的直方图$p'(l)$。要求求出灰度的变换关系，即灰度级之间的对应关系$l = T(k)$。例如$f(m, n)$中的第三级灰度变换为$g(m, n)$中的第二级，等等。

类似于式(4.1-14)和(4.1-15)，可以写出图像$f(m, n)$中具有$0, 1, \cdots, k$范围内的灰度级数的像素，占有整幅图像像素数的比例为：

$$F(k) = \sum_{i=0}^{k} p(i),\ k = 0, 1, \cdots, K-1 \tag{4.1-20}$$

同样，对图像$g(m, n)$有：

$$F'(l) = \sum_{j=0}^{l} p'(j),\ l = 0, 1, \cdots, K'-1 \tag{4.1-21}$$

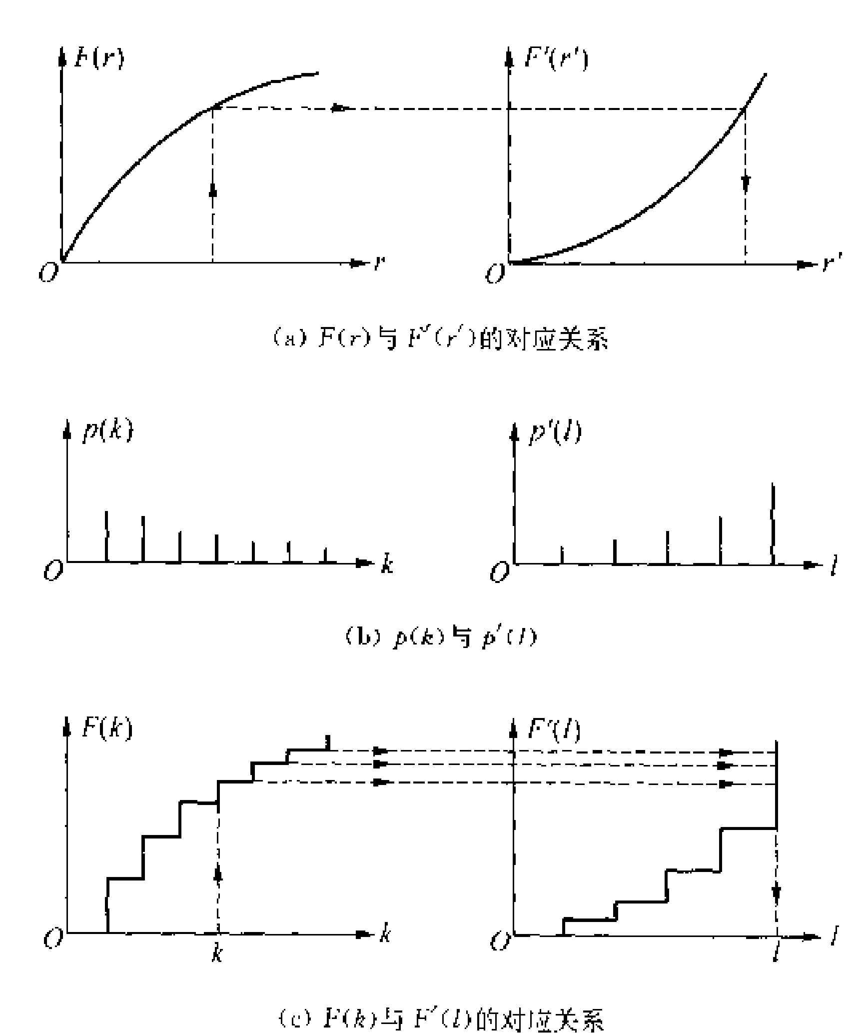
在连续图像的场合，可以建立起由式(4.1-16)所确定的变换关系。这是因为r和r'取的是实数，用等号连接$F(r)$和$F'(r')$是合理的，其对应关系如图4.10(a)所示。

然而在数字图像的情况下，k和l表达的是灰度级数，取的是整数值，因此$F(k)$和$F'(l)$都不是连续的，见图4.10(c)。对于某个$F(k)$的值并不一定能找到一个$F'(l)$与之相等，用等号连接它们来寻找l与k的关系将招致求解失败，合理的方法是用不等式把它们的关系确定下来，即：

$$F'(l-1) < F(k) \leqslant F'(l) \tag{4.1-22}$$

这就是说当 $f(m, n)$中第 k 级灰度，对应 $g(m, n)$中的第 l 和 $l-1$ 级灰度，使三者相应的比例值 $F'(l-1)$、$F(k)$、$F'(l)$满足(4.1-22)式，即 $F(k)$的值落入子区间 $(F'(l-1), F'(l)]$ 中时，便建立了 k 和 l 之间的对应关系。

由于 $l=0$ 时有 $l-1=-1$，故特别规定 $F'(-1)=0$。



(a) $F(r)$与 $F'(r')$的对应关系

(b) $p(k)$与 $p'(l)$

(c) $F(k)$与 $F'(l)$的对应关系

图 4.10 $f(m, n)$图像灰度的非线性变换

从图 4.10(c)可见，子区间 $(F'(l-1), F'(l)]$ 各段首尾衔接，也不重叠，故一个 $F(k)$值一定落入一个子区间中，对应一个 l 值。因此，从 $f(m, n)$的某个灰度级 k，求得 $g(m, n)$的灰度级 l 时，l 是唯一的；同时由于可能会有多个 $F(k)$值落入同一子区间内，故可能会有多个 k 值与一个 l 值对应；此外，也可能没有任何 $F(k)$ 值落入某个子区间 $(F'(l-1), F'(l)]$ 内，亦即可能在 $g(m, n)$的某个灰度级中连一个像素也没有。

在实际应用中，数字图像灰度的非线性变换可以用"查表法"来实现。可以建一个灰度变换表，用数组 Lut[0, $K-1$]来表示，按所要求的灰度变化规律给数组元素

Lut[k]赋值，Lut[k]中存放原灰度值为 k 经变换后的灰度值，即 Lut[k] $= l$，$l = T(k)$。这样，灰度变换就成了逐像素"查表"的简单过程。在大多数实用系统中，经常从"帧存"中读出 $f(m, n)$值，经"输出查找表 Lut"变换后再进行 D/A 转换为模拟信号后显示出来。这时做灰度变换根本不需要改变帧存的值，而只需改变 Lut 的内容。

例 1 设 $f(m, n)$有 $K = 16$ 级灰度，$g(m, n)$有 $K' = 9$ 级灰度，各自的直方图 $p(k)(k = 0, 1, \cdots, 15)$、$p'(l)(l = 0, 1, \cdots, 8)$ 如表 4.1 和 4.2 所示。希望在灰度变换后得到的 $g(m, n)$具有 $p'(l)$那样的直方图。求灰度级 k、l 的对应关系，并求 $g(m, n)$实际的直方图 $Q(l)$。

解：求 $l = T(k)$ 对应关系的过程可以由计算加填表来完成，见表 4.1 和表 4.2。

(1) 根据式(4.1-20)、(4.1-21)，由 $f(m, n)$的直方图 $p(k)$及 $g(m, n)$希望达到的直方图，计算 $F(k)$和 $F'(l)$。

(2) 由式(4.1-22)，观察每个 $F(k)$落入哪一个子区间 $(F'(l-1), F'(l)]$，以确定 k、l 的对应关系。从表 4.1 中可以看到 $f(m, n)$的第 0、1、2、3 级灰度同时对应 $g(m, n)$的第 0 级，而 $g(m, n)$的第 1 级则没有相应的原始灰度级与之对应。

表 4.1 灰度非线性变换举例附表一

原图灰度级 k	0	1	2	3	4	5	6	7
直方图 $p(k)$	0.03	0	0.06	0.10	0.20	0.11	0	0
比例值 $F(k)$	0.03	0.03	0.09	0.19	0.39	0.50	0.50	0.50
对应灰度级 l	0	0	0	0	2	3	3	3
原图灰度级 k	8	9	10	11	12	13	14	15
直方图 $p(k)$	0	0.03	0	0.06	0.10	0.20	0.11	0
比例值 $F(k)$	0.50	0.53	0.53	0.59	0.69	0.89	1.00	1.00
对应灰度级 l	3	5	5	6	7	8	8	8

表 4.2 灰度非线性变换举例附表二

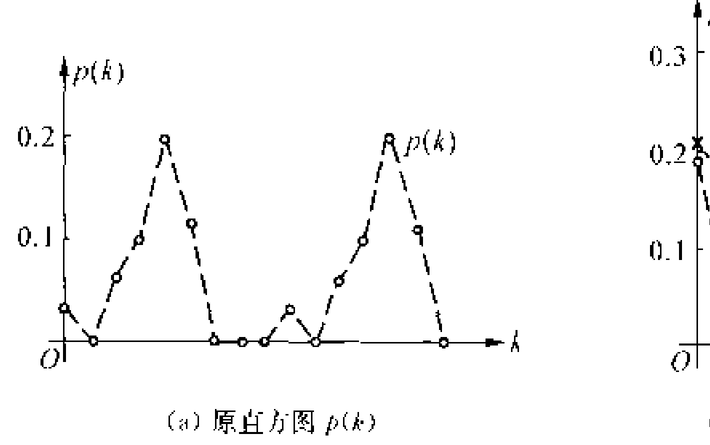
修改后灰度级 l	0	1	2	3	4	5	6	7	8
直方图 $p'(l)$	0.20	0.15	0.10	0.05	0	0.05	0.10	0.15	0.20
比例值 $F'(l)$	0.20	0.35	0.45	0.50	0.50	0.55	0.65	0.80	1.00
实际直方图 $Q(l)$	0.19	0	0.20	0.11	0	0.03	0.06	0.10	0.31

(3) 计算变换后实际得到的 $g(m, n)$的直方图 $Q(l)$。

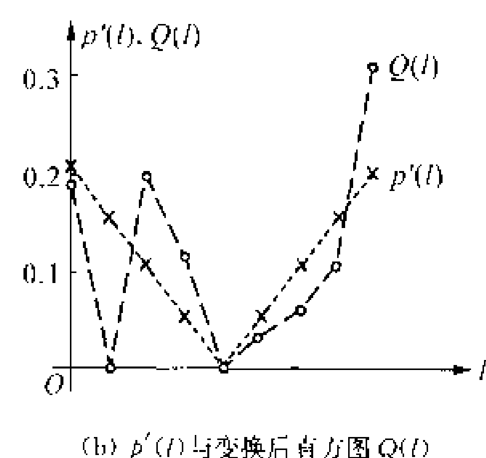
图 4.11 示出了本例中的直方图 $p(k)$、$p'(l)$、$Q(l)$。

从上例可见，虽然目标是将直方图由 $p(k)$变换为 $p'(l)$，但实际得到的却是接近 $p'(l)$的 $Q(l)$，见图 4.11(b)。出现这种现象是正常的，在目前的条件下很难避

免。这是因为,离散灰度情况与连续灰度情况有着根本差别。在离散情况下,原直方图中最多只有 K 条与 K 个灰度值对应的表示概率大小的垂直线,作灰度变换时只不过把这些垂直线的水平位置进行平移和重新分布而已。由于原图中同一灰度值 k 的所有像素都变为灰度 l,不会只有一部分变为 l,而另一部分变为其他灰度,所以原直方图中每条线的“高度”不能随便改变。当多个不同 k 值的垂直线移到同一个 l 时,才发生灰度合并,与 l 对应的线的高度变为 $\sum p(k)$。若变换前后灰度级数相同,即 $K = K'$,则若有的地方发生灰度合并,就必有其他地方留出“空档”,即存在无任何像素的灰度级 l。综上所述,在离散灰度情况下发生 $Q(l) \neq p'(l)$ 的现象是正常的。为使 $Q(l)$尽量接近于 $p'(l)$,在变换方法上所能做的变通,不过是当有多个$F(k)$同时落在 $F'(l-1)$~$F'(l)$区间内时,可以在 $k_i \to l$ 还是 $k_i \to l-1$ 之间作些调整。例如,若 $F(k_1)$、$F(k_2)$和 $F(k_3)$同时落在 $F'(l-1)$~$F'(l)$区间内,而 $F(k_1)$靠近 $F'(l-1)$,$F(k_2)$和 $F(k_3)$更接近于 $F'(l)$,则可以将 $k_1 \to l-1$,而 k_2,$k_3 \to l$。



(a) 原直方图 $p(k)$



(b) $p'(l)$与变换后直方图 $Q(l)$

图 4.11 灰度非线性变换例子

前面从连续灰度级情况分析灰度增强,会达到理想的处理效果,但实际用计算机对数字图像做增强时,会遇到不少具体问题。产生这些问题的原因,不在于处理方法不好,而是从理论上就是有问题的。这些问题都与数字图像中有效灰度级,即概率值非零的灰度级太少有关。下面通过两个例子来分别说明离散情况下进行灰度增强时可能出现的伪轮廓现象和灰度级合并现象。

例 2 伪轮廓现象。若原图曝光严重不足,直方图如图 4.12 所示,要求通过灰度增强来改善图像质量。

解:对于连续图像的情况,处理起来很方便,只要用前面所介绍的灰度线性变换把暗区拉大就可以改善图像质量。

而数字图像的灰度级经过了分层量化,从图 4.12 中可见,该图像中只有灰度值分别为 10、11、12、13、14、15 共六个有效灰度级。如果用变换 $l = T(k)$ 来进行灰度变换,使灰度值发生如下变换:10→10、11→30、12→50、13→70、14→90、

15→110。则变换后直方图如图 4.13 所示。

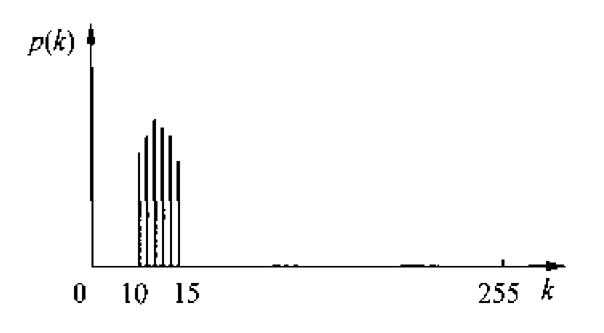


图 4.12 原图曝光不足对应的直方图

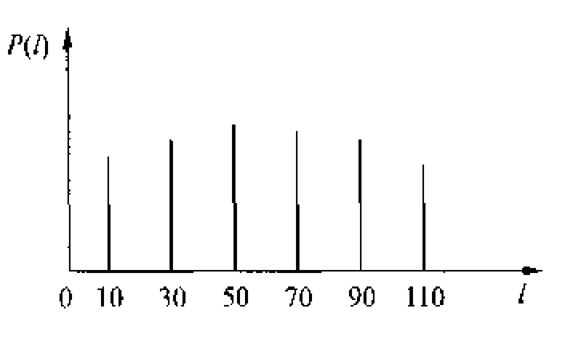
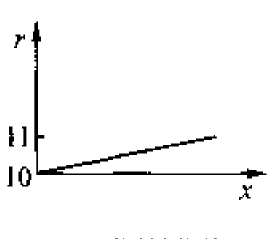
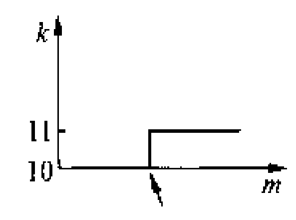


图 4.13 变换后的直方图

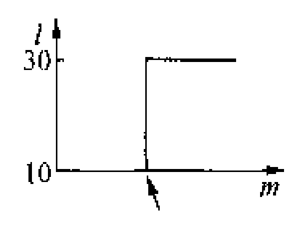
从图 4.13 可以看出，"非零"即"有效"的灰度级仍然只有六个，虽然灰度级在变换后被"排开"，人眼能够分清它们之间的区别，但是会看到"伪轮廓"，如图 4.14 所示。



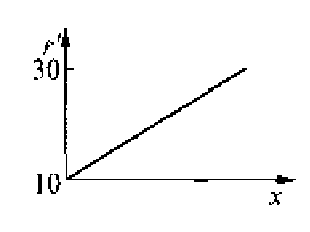
(a) 原图量化前



(b) 量化后的数字图像



(c) 数字图像增强后



(d) 连续图像增强后

图 4.14 伪轮廓现象示意图

图 4.14 中横坐标为图像空间位置坐标 x(取样前)或 m(取样后)，纵坐标为灰度值 r(量化前)或 k(量化后)。图 4.14(a)中显示出图像 $f(x, y)$中有一片区域的灰度值从"10"连续变化到"11"，由于在该区域内并没有出现灰度值的突变，所以该区域内没有边缘(轮廓)存在。若进行灰度变换，拉大灰度值的取值范围，例如把灰度值从"10"连续变化到"11"变换为灰度值从"10"连续变化到"30"，则变换效果如图 4.14(d)所示。由于变换后该区域中没有出现灰度值的突变，所以仍然没有边缘存在。当 $f(x, y)$量化为 $f(m, n)$后，原来灰度值从"10"连续变化到"11"的区域对应于"10"和"11"两个灰度值，如图 4.14(b)所示，在该区域的中间位置会有一个灰度值从"10"变为"11"的突变。此时由于只跳变了一个灰度级，人眼不会发现这个灰度值的变化，故该区域看上去仍然没有边缘(轮廓)存在。而当进行了灰度变换，将"10"变为"10"、"11"变为"30"后，在该区域的中间位置就出现了灰度值从"10"变为"30"的突变，如图 4.14(c)所示。此时由于跳变了 20 个灰度级，人眼会在灰度突变处看到一条边，即轮廓。由于这个轮廓是原图中所没有的，故称之为"伪轮廓"。

事实上，当数字图像的有效灰度级总数小于 16 或 32 时，在增强处理后都会出现伪轮廓现象。对于伪轮廓现象的解决办法有两个，一是用模拟增强，即用模拟放

大器来实现灰度变换 $r' = T(r)$；二是增加灰度量化精度，例如上例中若把 8 bit 精度增加为 12 bit 精度，则有效灰度级可从 10～15 的 6 级增加为 $6\times2^4 = 6\times16 = 96$ 级，此时再做增强就不会产生伪轮廓现象。

例 3 灰度级合并现象。若进行如图 4.15 所示的灰度变换，分析变换结果。

解：对连续图像进行如图 4.15 中所示的灰度变换时不会出现任何问题。但是，在离散灰度情况下，由于 $[r_a, r_c]$ 范围内的灰度级经变换后，有效灰度级级数不会增加，而是补"0"；而 $(r_c, r_b]$ 范围内的灰度变换是 2 级并成 1 级，故总起来看，变换后有效灰度级数只会变少，而不会增加。所以，图像增强并不会增加图像的信息量，而只可能减少图像的信息量。但由于增强后图像的表示方法更符合人眼视觉特性，例如使图像中原来由于不符合人眼视觉特性而无法看清的部分经增强后可以看清楚，从而使人们认为图像增强可明显改善图像质量。

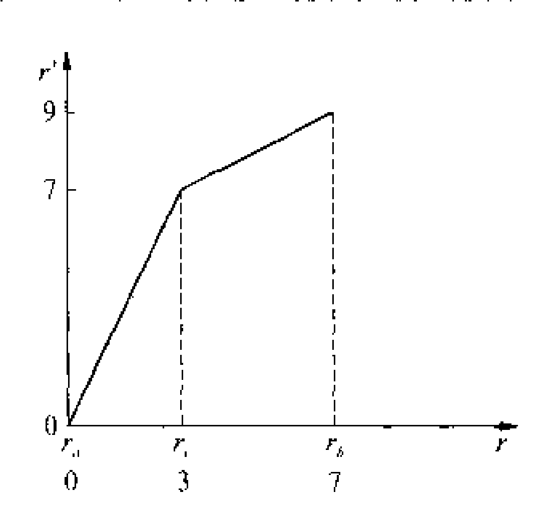


图 4.15 出现灰度级合并现象的灰度变换图

4.1.4 直方图均衡和直方图规定化

当要求灰度变换后的直方图 $p'(l) = \frac{1}{K'}$ $(l = 0, 1, \cdots, K'-1)$ 时，这种非线性变换又称为直方图均衡。在指定某一形状的目标直方图时，这种非线性变换又称为直方图规定化。直方图均衡其实就是直方图规定化的一个特例。在 4.1.3 小节中，图 4.8、图 4.10 和例 1 中的非线性变换都是直方图规定化，而图 4.9 中的非线性变换则是直方图均衡。

这里从理论上对直方图均衡和直方图规定化进行分析。

1. 灰度连续变化情况的直方图均衡

直方图均衡就是找出一种灰度变换来使变换后图像的直方图是"均匀分布"的，即 $p(r)$ 或 $p(k)$ 为一常数。一般来说，直方图均衡中的灰度变换是非线性的。

做直方图均衡的理论依据是最大熵原理。

在灰度连续变换情况下，定义熵

$$H_c = -\int_{r_{\min}}^{r_{\max}} p(r)\log p(r)\mathrm{d}r \tag{4.1-23}$$

可以证明，当 $p(r)$ 为常数时，即直方图均衡时，熵最大，即可使视觉系统获取

最大信息量。这就是最大熵原理。

证明：若要求 $p(r)$，使 H_c 最大，且满足约束条件 $\int_{r_{min}}^{r_{max}} p(r)\mathrm{d}r = 1$。

用拉格朗日乘子法，令目标函数

$$J = -\int_{r_{min}}^{r_{max}} p(r)\log(r)\mathrm{d}r + \lambda\left(\int_{r_{min}}^{r_{max}} p(r)\mathrm{d}r - 1\right) \tag{4.1-24}$$

令 $\frac{\partial J}{\partial p(r)} = 0$，求 $p(r)$。

因为
$$\begin{aligned}\frac{\partial J}{\partial p(r)} &= -\int_{r_{min}}^{r_{max}} \left\{\frac{\partial}{\partial p(r)}[p(r)\log p(r)] - \lambda\frac{\partial p(r)}{\partial p(r)}\right\} \\ &= -\int_{r_{min}}^{r_{max}} \left\{p(r)\cdot\frac{1}{p(r)} + 1\cdot\log p(r) - \lambda\right\}\mathrm{d}r \\ &= -\int_{r_{min}}^{r_{max}} \{\log p(r) + 1 - \lambda\}\mathrm{d}r \end{aligned} \tag{4.1-25}$$

由 $\log p(r) + 1 - \lambda = 0$，解得 $p(r) = 2^{\lambda-1} =$ 常数，即 $p(r)$ 与灰度 r 无关。

由上述过程可见，当 $p(r)=$ 常数时，即直方图均衡时，可使视觉系统获取最大信息量。

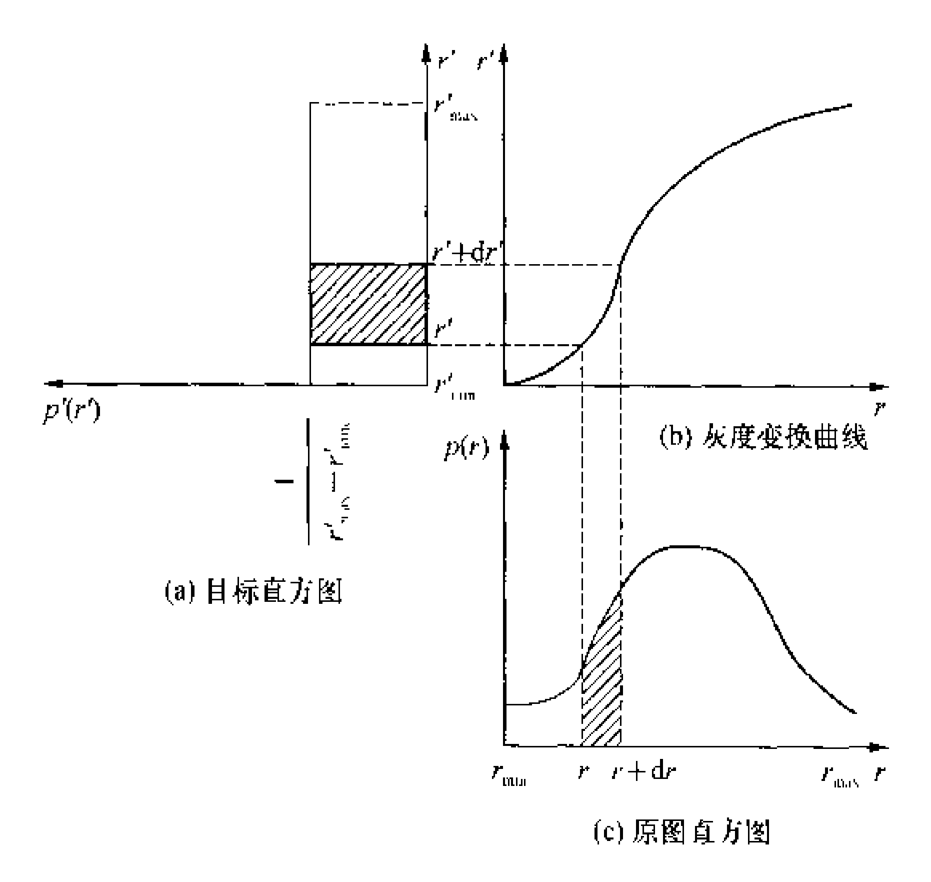


图 4.16 灰度连续变化情况下的直方图均衡

从图 4.16 可以看出，为使变换后的直方图是均匀分布的，可以在 r 小、$p(r)$ 小

时压缩直方图，使分布密度变大；在 r 中等、$p(r)$ 大时扩张直方图，使分布密度变小；在 r 大、$p(r)$ 小时压缩直方图，使分布密度变大。

这里用概率论中的常用方法：由密度函数 $p(r)$ 求分布函数 $F(r)$。定义分布函数为灰度 $\leqslant r$ 时的概率，即 $F(r)=\int_{r_{min}}^{r}p(w)\mathrm{d}w$。可以证明：当 $r'=T(r)=F(r)$ 时，即当灰度变换规律等于分布函数 $F(r)$ 时，就实现了直方图均衡。

证明：

由式(4.1-13)可知，原图中灰度值 $\leqslant r$ 的面积(百分比) $F(r)=\int_{r_{min}}^{r}p(w)\mathrm{d}w$ 等于变换后图像中的灰度 $\leqslant r'$ 的面积(百分比) $F'(r')=\int_{r'_{min}}^{r'}p'(w')\mathrm{d}w'$。

即不论 $r'=T(r)$ 的具体形状，只要 r' 与 r 对应，则分布函数 $F(r)=F'(r')$，即

$$\int_{r_{min}}^{r}p(w)\mathrm{d}w=\int_{r'_{min}}^{r'}p'(w')\mathrm{d}w' \tag{4.1-26}$$

为便于说明问题，将灰度值归一化为 0～1，则

$$p'(r')=\frac{1}{r'_{max}-r'_{min}}=\frac{1}{1-0}=1 \tag{4.1-27}$$

又因为 $p(r)\mathrm{d}r=p'(r')\mathrm{d}r'$

所以

$$\mathrm{d}r'=p(r)\mathrm{d}r \tag{4.1-28}$$

即

$$r'=\int_{0}^{r}p(w)\mathrm{d}w=F(r) \tag{4.1-29}$$

因此，为进行直方图均衡，在灰度连续情况下，灰度变换曲线 $T(r)$ 正好选为原图的分布函数 $F(r)$ 的形状。

2. 灰度连续变化情况的直方图规定化

有时候，为了各种不同应用目的，要求不同形状的目标直方图，这就是直方图规定化。

在灰度连续变化情况下，直方图规定化可描述如下：

已知原图直方图 $p(r)$，求一种线性变换 $r'=T(r)$，使变换后的直方图 $p'(r')$ 的形状等于事先指定的形状。

如图 4.17 所示，可以把直方图规定化看成以下几步：

(1) 先作原图直方图均衡，即由图 4.17(c)将原图直方图 $r\sim p(r)$ 经直方图均衡变为 $r''\sim p''(r'')$，故 $r''=F(r)$，如图 4.17(a)所示。

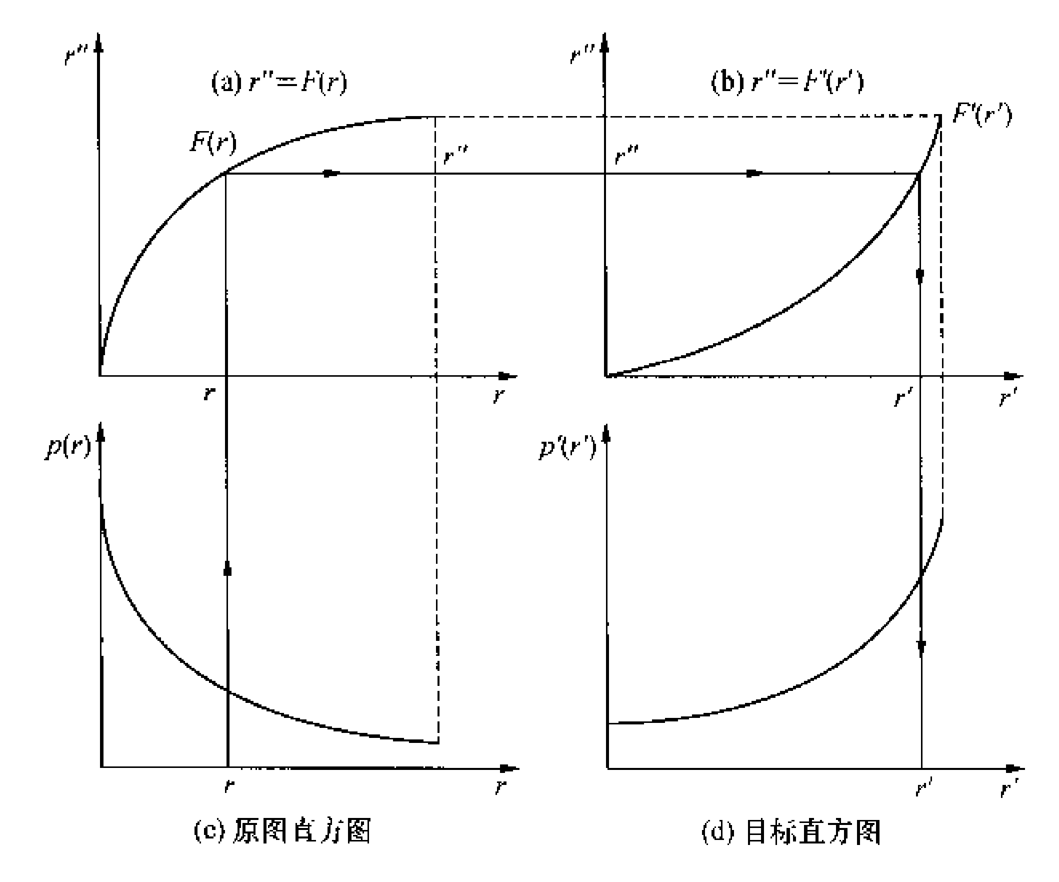


图 4.17 灰度连续变化情况下的直方图规定化

(2) 再作目标直方图的反均衡，即若由图 4.17(d)将目标直方图 $r'\sim p'(r')$经直方图均衡为 $r''\sim p''(r'')$时需作变换 $r''=F'(r')$，反均衡就是反变换，即从均匀分布的 $p''(r'')$经反变换变为 $p'(r')$，故 $r'=F'^{-1}(r'')$，如图 4.17(b)所示。

(3) 由(1)、(2)可得：$r'=F'^{-1}(F(r))$。

从上述分析可以得出，直方图规定化的过程为：

(1) 由原图直方图 $p(r)$求出 $F(r)$，$r''=F(r)$；

(2) 由目标直方图 $p'(r')$求出 $F'(r')$，$r''=F'(r')$；

(3) 由 $F'(r')$求其反变换 $r'=F'^{-1}(r'')$；

(4) 直方图规定化所需灰度变换曲线为：$r'=F'^{-1}(F(r))$。

3. 灰度离散化情况的直方图均衡和直方图规定化

在灰度离散化情况下进行直方图均衡和直方图规定化时，前面对连续图像所作的所有分析都适用。

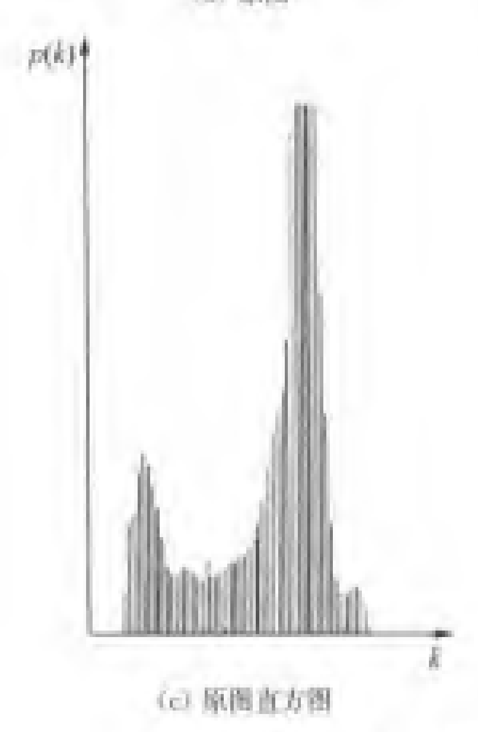
在灰度离散化情况下，直方图均衡处理只是拉大了图像的对比度，使人眼看起来更清晰些，但是由于发生了灰度级合并现象，图像的灰度层次却减少了。图 4.18 给出了一幅数字图像进行直方图均衡后的视觉效果。



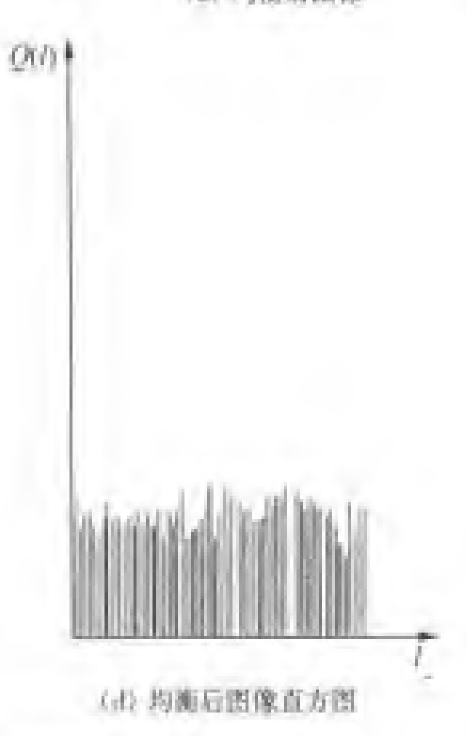
(a) 原图



(b) 均衡后图像



(c) 原图直方图



(d) 均衡后图像直方图

图 4.18 直方图均衡举例

4.2 空间域图像滤波

上一节中介绍的灰度增强是一种点变换的处理方法，而在数字图像处理中，最典型的方法是空间域卷积，或称"开窗"运算或"模板"运算。例如已知数字图像 $f(m, n)$ 和空间域处理模板(Mask)为：

$$[\boldsymbol{M}] = \begin{bmatrix} m_{11} & m_{12} & m_{13} \\ m_{21} & m_{22} & m_{23} \\ m_{31} & m_{32} & m_{33} \end{bmatrix} \tag{4.2-1}$$

则模板运算处理对应于

$$
\begin{aligned}
g(m,n) = {} & f(m-1,n-1)m_{11} + f(m-1,n)m_{12} + \cdots \\
& + f(m+1,n+1)m_{33}
\end{aligned}
\tag{4.2-2}
$$

即处理结果只与模板(9个系数)和 $f(m,n)$ 及其8个空间邻点的值有关。

如果令

$$
[\boldsymbol{h}(m,n)] = \begin{bmatrix} m_{33} & m_{32} & m_{31} \\ m_{23} & m_{22} & m_{21} \\ m_{13} & m_{12} & m_{11} \end{bmatrix}
\tag{4.2-3}
$$

则可写成离散卷积形式

$$
g(m,n) = f(m,n) * [\boldsymbol{h}(m,n)]
\tag{4.2-4}
$$

用不同的模板可以直接在空间域进行边缘增强、噪声平滑和边缘检测等处理。本节从空间域处理的角度来讨论图像的锐化和平滑，下一节将从频率域处理的角度来讨论图像的锐化和平滑。

4.2.1　空间域图像锐化

摄像系统的聚焦不好或者信道的通带过窄，都会造成图像中目标轮廓的模糊。图像锐化可以加重图像的轮廓。实验表明目标轮廓加重的图像常比精确再现的图像更令人满意。图4.19给出一幅目标轮廓模糊的图像在锐化前后的效果。



(a) 模糊图像



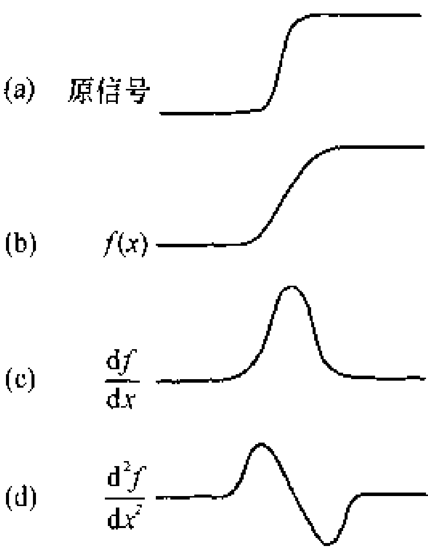
(b) 锐化后图像

图4.19　图像锐化举例

图 4.20 示出了一维波形锐化的过程：(a)为原信号；(b)为模糊以后的波形，它变成一个缓慢上升的波形，设它是 $f(x)$；(c)和(d)分别是(b)的一阶和二阶微商波形；(e)为锐化处理以后的波形，处理的表达式是：

$$g(x) = f(x) - \frac{d^2 f(x)}{dx^2} \qquad (4.2-5)$$

经(4.2-5)式处理后，即缓慢上升的波形减去其二次微商后，所得的波形 $g(x)$ 上升较快，并有“过冲”出现，这就使得信号的边缘突出，达到了锐化的目的。



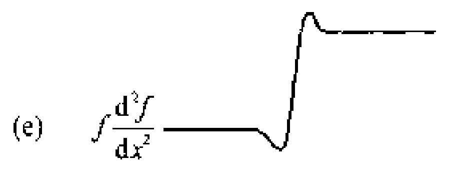


图 4.20 一维波形的锐化

在二维情况下，要求找到一种算法使走向不同的轮廓能够在垂直于边缘的方向上具有类似图 4.20(e)那样的锐化效应。用拉普拉斯算子实现的算法可以实现上述目的。拉普拉斯算子及其对 $f(x, y)$ 的作用是：

$$\nabla^2 = \frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2}, \quad \nabla^2 f = \frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2} \qquad (4.2-6)$$

可以证明，在连续情况下，这是一种各向同性的运算，从而可以满足不同走向的轮廓锐化的要求。

由一维的锐化表达式(4.2-5)，可写出二维表达式为：

$$g(x, y) = f(x, y) - \alpha \nabla^2 f(x, y) \qquad (4.2-7)$$

式中：α 因子用于调节锐化程度。

对于数字图像，可用差分来近似连续系统中的微商，其对应关系见表 4.3。

表 4.3 微商与差分的对应关系

连续		离散	
连续图像 $f(x, y)$		数字图像 $f(m, n)$	
微商		前向差分	后向差分
一阶	$f'_x(x, y)$	$f(m+1, n) - f(m, n)$	$f(m, n) - f(m-1, n)$
	$f'_y(x, y)$	$f(m, n+1) - f(m, n)$	$f(m, n) - f(m, n-1)$

（续表）

连续		离散	
连续图像 $f(x, y)$		数字图像 $f(m, n)$	
微商		前向差分	后向差分
二阶	$f''_{xx}(x, y)$	$f(m+1, n)+f(m-1, n)-2f(m, n)$	
	$f''_{yy}(x, y)$	$f(m, n+1)+f(m, n-1)-2f(m, n)$	

因此，数字图像的锐化表达式可写成：

$$
\begin{aligned}
g(m, n) &= f(m, n)-\alpha[f(m+1, n)+f(m-1, n)+f(m, n+1) \\
&\quad +f(m, n-1)-4f(m, n)] \\
&= [1+4\alpha]f(m, n)-\alpha\sum_{四}\sum_{邻} f(i, j)
\end{aligned}
\tag{4.2-8}
$$

其中求和中的四邻是指位于(m, n)点上下左右的四个邻点，即$(m\pm 1, n)$和$(m, n\pm 1)$。$\alpha>0$，它是一个可供选择的参数。α越大，图像锐化得越厉害。当$f(m, n)$与其四邻的灰度值都相等时，即图像的这一部分为常数，没有轮廓存在，由式(4.2-8)处理后还是原来的值$f(m, n)$，这时没有锐化作用。

应该指出的是，在连续情况下拉普拉斯算子是各向同性的，但是对数字图像，这一锐化算子不再严格保持各向同性的性质。这是因为在离散情况下用差分来近似微商的缘故。

式(4.2-8)可以用下面的模板来表示：

$$
[\boldsymbol{M}_1]=\begin{bmatrix} 0 & -\alpha & 0 \\ -\alpha & 1+4\alpha & -\alpha \\ 0 & -\alpha & 0 \end{bmatrix}
\tag{4.2-9}
$$

该模板是可以移动的，见图4.21。把模板的中心对准要处理的点(m, n)，模板上的值就是对应各点灰度值的加权值，把这些灰度值加权求和后，就得到了处理后的

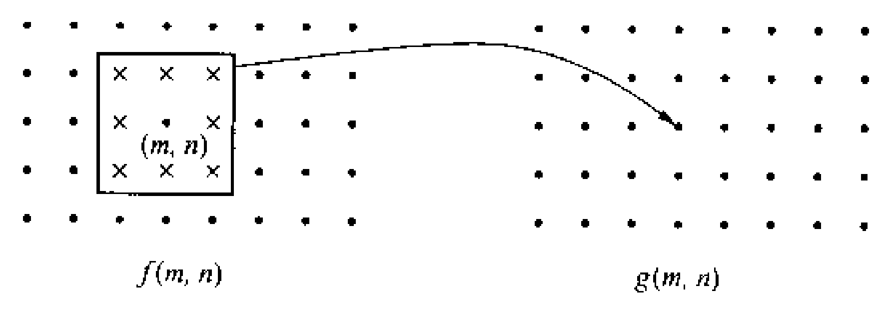


图4.21 用模板作图像处理

灰度值 $g(m, n)$。这个模板有 3 行 3 列,常称为 3×3 模板。

如果把(4.2-8)式中的四邻改写成八邻,可得到:

$$g(m, n) = (1+8\alpha)f(m, n) - \alpha \sum_{\text{八}} \sum_{\text{邻}} f(i, j) \tag{4.2-10}$$

对应模板为:

$$[\boldsymbol{M}_2] = \begin{bmatrix} -\alpha & -\alpha & -\alpha \\ -\alpha & 1+8\alpha & -\alpha \\ -\alpha & -\alpha & -\alpha \end{bmatrix} \tag{4.2-11}$$

这也是一种轮廓锐化的运算。可以注意到模板$[\boldsymbol{M}_1]$和$[\boldsymbol{M}_2]$中的 9 个系数相加等于 1,因为对一个常数图像处理时,图像应当不变;对一般图像处理后,整幅图像的平均值也可不变。$[\boldsymbol{M}_1]$和$[\boldsymbol{M}_2]$是图像锐化中最普遍使用的公式,根据它们的基本原理,可以发展出一些更复杂的公式。

模板$[\boldsymbol{M}_2]$可以分解为下面 9 个模板,其中$[\boldsymbol{M}_{2-0}]$处理图像后仍得到原图,而$[\boldsymbol{M}_{2-1}]$至$[\boldsymbol{M}_{2-8}]$是差分模板:

$$\underset{[\boldsymbol{M}_{2-0}]}{\begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}} \quad \underset{[\boldsymbol{M}_{2-1}]}{\begin{bmatrix} 0 & 0 & 0 \\ 0 & \alpha & -\alpha \\ 0 & 0 & 0 \end{bmatrix}} \quad \underset{[\boldsymbol{M}_{2-3}]}{\begin{bmatrix} 0 & -\alpha & 0 \\ 0 & \alpha & 0 \\ 0 & 0 & 0 \end{bmatrix}} \quad \underset{[\boldsymbol{M}_{2-5}]}{\begin{bmatrix} 0 & 0 & 0 \\ -\alpha & \alpha & 0 \\ 0 & 0 & 0 \end{bmatrix}} \quad \underset{[\boldsymbol{M}_{2-7}]}{\begin{bmatrix} 0 & 0 & 0 \\ 0 & \alpha & 0 \\ 0 & -\alpha & 0 \end{bmatrix}}$$

$$\underset{[\boldsymbol{M}_{2-2}]}{\begin{bmatrix} 0 & 0 & -\alpha \\ 0 & \alpha & 0 \\ 0 & 0 & 0 \end{bmatrix}} \quad \underset{[\boldsymbol{M}_{2-4}]}{\begin{bmatrix} -\alpha & 0 & 0 \\ 0 & \alpha & 0 \\ 0 & 0 & 0 \end{bmatrix}} \quad \underset{[\boldsymbol{M}_{2-6}]}{\begin{bmatrix} 0 & 0 & 0 \\ 0 & \alpha & 0 \\ -\alpha & 0 & 0 \end{bmatrix}} \quad \underset{[\boldsymbol{M}_{2-8}]}{\begin{bmatrix} 0 & 0 & 0 \\ 0 & \alpha & 0 \\ 0 & 0 & -\alpha \end{bmatrix}}$$

而$[\boldsymbol{M}_1]$模板可看成是其中$[\boldsymbol{M}_{2-0}]$、$[\boldsymbol{M}_{2-1}]$、$[\boldsymbol{M}_{2-3}]$、$[\boldsymbol{M}_{2-5}]$、$[\boldsymbol{M}_{2-7}]$等 5 个模板的合成。

模板的分解表明,锐化模板$[\boldsymbol{M}_1]$或$[\boldsymbol{M}_2]$可以理解为图像本身和按不同方向差分(微商)后的图像的总和。这些差分图像实际上是各个不同方向上的轮廓。因此图像的锐化就相当于在原图上加一个轮廓。

由于$[\boldsymbol{M}_1]$和$[\boldsymbol{M}_2]$模板简单、效果明显,因此获得广泛的应用。然而对于图像上原来已有的(也是不可避免的)噪声,同样具有增强的效果,一般不宜进行迭代运算。

这种模板对孤立点及孤立线条的增强作用比对图像中某一区域边缘的增强作用要明显得多,在应用中应当十分注意。

4.2.2 空间域图像平滑

一幅图像可能受到各种噪声源的干扰,包括光电转换过程中的噪声、照片颗粒

噪声和信息传输中的误差等。图像的平滑处理可以减小这些噪声。本节从空间域的角度来讨论图像的平滑。

1. 邻域平均法

若原图上有很多噪声“麻点”，则可通过进行图像平滑来改善图像质量，用邻域平均法去噪是常用的方法。这是一种在空间域上对图像进行平滑处理的一种方法，易于实现，效果也比较好。

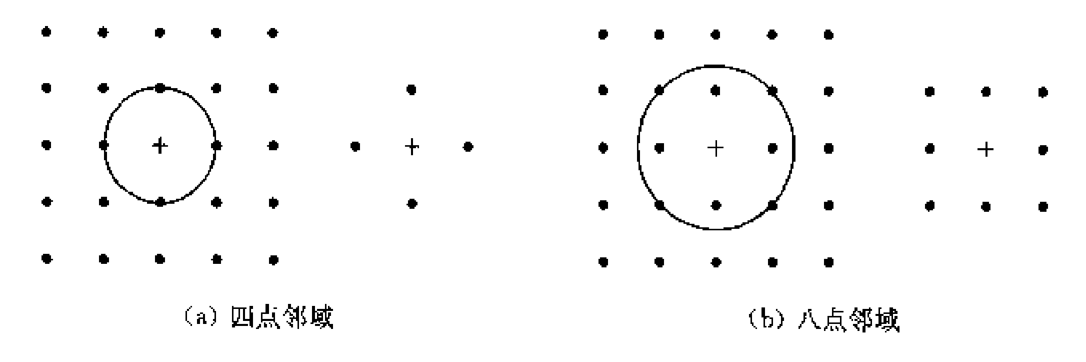
设噪声 $\eta(m, n)$是加性白噪声，其均值为0，方差（噪声功率）为 σ^2，而且噪声与图像 $f(m, n)$不相关。加有噪声的图像 $f'(m, n)$为：

$$f'(m, n) = f(m, n) + \eta(m, n) \tag{4.2-12}$$

经邻域平均法处理以后的图像 $g(m, n)$为：

$$\begin{aligned} g(m, n) &= \frac{1}{N}\sum_{(i,j)}\sum_{\in S} f'(i, j) = \frac{1}{N}\sum_{(i,j)}\sum_{\in S}[f(i, j) + \eta(i, j)] \\ &= \frac{1}{N}\sum_{(i,j)}\sum_{\in S} f(i, j) + \frac{1}{N}\sum_{(i,j)}\sum_{\in S}\eta(i, j) \end{aligned} \tag{4.2-13}$$

其中 S 是(m, n)点的邻域内的点集，图4.22是这种点集的两个例子。



(a) 四点邻域　　(b) 八点邻域

图4.22 (m, n)点的两种邻域

除了对噪声有上述假定之外，该算法还基于这样一种假设：图像是由许多灰度值相近的小块组成。这个假设大体上反映了许多图像的结构特征。式(4.2-13)表达的算法是由某像素邻域内的各点灰度值的平均值来代替该像素原来的灰度值。

设 $E\{\cdot\}$和 $D\{\cdot\}$表示求统计平均和方差的运算，则：

$$E\left\{\frac{1}{N}\sum_{(i,j)}\sum_{\in S}\eta(i, j)\right\} = \frac{1}{N}\sum_{(i,j)}\sum_{\in S}E\{\eta(i, j)\} = 0 \tag{4.2-14}$$

$$D\left\{\frac{1}{N}\sum_{(i,j)}\sum_{\in S}\eta(i, j)\right\} = \frac{1}{N^2}\sum_{(i,j)}\sum_{\in S}D\{\eta(i, j)\} = \frac{1}{N}\sigma^2 \tag{4.2-15}$$

式(4.2-14)表明邻域平均之后，残余噪声的平均值仍为0，而方差则由原来的σ^2下

降为$\frac{1}{N}\sigma^2$。但是图像则由$f(m,n)$变为$\frac{1}{N}\sum\sum_{(i,j)\in S}f(i,j)$。这个变化引起了$g(m,n)$的失真，具体表现在图像中目标物的轮廓或细节变模糊了。邻域平均法的处理效果与前一小节中由拉普拉斯算子等所确定的图像锐化所处理的效果正好相反。后者可使目标轮廓或细节变清晰，同时对噪声也有增强作用。

可用模板反映邻域平均算法的特征。对于图 4.24 中的小圆邻域和大圆邻域，可分别由下述模板表征：

$$[\boldsymbol{M}_5]=\begin{bmatrix}0 & \frac{1}{5} & 0\\ \frac{1}{5} & \frac{1}{5} & \frac{1}{5}\\ 0 & \frac{1}{5} & 0\end{bmatrix}=\frac{1}{5}\begin{bmatrix}0 & 1 & 0\\ 1 & 1 & 1\\ 0 & 1 & 0\end{bmatrix} \tag{4.2-16}$$

$$[\boldsymbol{M}_6]=\begin{bmatrix}\frac{1}{9} & \frac{1}{9} & \frac{1}{9}\\ \frac{1}{9} & \frac{1}{9} & \frac{1}{9}\\ \frac{1}{9} & \frac{1}{9} & \frac{1}{9}\end{bmatrix}=\frac{1}{9}\begin{bmatrix}1 & 1 & 1\\ 1 & 1 & 1\\ 1 & 1 & 1\end{bmatrix} \tag{4.2-17}$$

模板沿水平和垂直两个方向逐点移动，从而平滑了整幅图像。与 4.2.1 小节中的锐化模板一样，平滑模板内各系数和为 1，用这样的模板处理常数图像时，图像没有变化；对一般图像处理后，整幅图像灰度的平均值可以不变。

图 4.23 中显示出一幅图像在噪声污染后分别用式(4.2-16)和式(4.2-17)进行平滑处理后的效果。

有时为了突出(m,n)点本身的重要性，以便在一定程度上减少失真，可采用下述模板：

$$[\boldsymbol{M}_7]=\frac{1}{10}\begin{bmatrix}1 & 1 & 1\\ 1 & 2 & 1\\ 1 & 1 & 1\end{bmatrix},\ [\boldsymbol{M}_8]=\frac{1}{16}\begin{bmatrix}1 & 2 & 1\\ 2 & 4 & 2\\ 1 & 2 & 1\end{bmatrix}$$

本节中介绍的模板都是 3×3 大小的模板。如果模板大，去噪效果会更明显，但是计算复杂，所涉及的像素多而容易把细节抹去，造成模糊。在 3×3 模板去噪声效果不明显时，可以重复迭代几次，并监视中间结果。

由于邻域平均法的缺点是使图中的边缘变模糊，为解决平滑噪声与保持边缘的矛盾，有各种边缘保持的噪声平滑法被提出来，分别介绍如下。



(a) 原图



(b) 加椒盐噪声后的图像



(c) 用式(4.2-16)平滑后的图像



(d) 用式(4.2-17)平滑后的图像

图4.23 图像平滑举例

2. 带门限的邻域平均法

这一算法的数学表达式如下：

$$g(m,n)=\begin{cases}\dfrac{1}{N}\sum\limits_{(i,j)\in S}f(i,j), & \left|f(m,n)-\dfrac{1}{N}\sum\limits_{(i,j)\in S}f(i,j)\right|>T\\ f(m,n), & \text{其他}\end{cases}$$

(4.2-18)

式中：S是点(m,n)的一个邻域；N是所取邻域内的像素数；T是预先设定的门

限值。

这种算法的思想是：当 $f(m, n)$ 与邻域均值的差别不大于门限的时候不做平均，以起到保护边界点的作用。这种算法可去除大的噪声点，对小的噪声则认为人眼可以容忍而不需去除。

3. 半邻域平均法

这种算法是先检测处理窗内有无边界，若没有边界，则进行 9 点平均；若有边界，则将中心点与同在一边的 5 点一起进行 6 点平均。从而实现不在处理窗内跨边界做平均，达到保持边缘的目的。

设在 3×3 邻域内图像为

$$
\begin{matrix}
A_3 & A_2 & A_1 \\
A_4 & P & A_0 \\
A_5 & A_6 & A_7
\end{matrix}
$$

可见 P 周围有 8 个邻点 $A_i(i=0, 1, \cdots, 7)$。半邻域平均法的具体算法为：

(1) 对 A_i 排序，灰度值较大的前 5 点构成 B 组，灰度值较小的后 3 点构成 A 组。

(2) 设定门限值 T。

(3) 求 A，B 两组的平均值 $\overline{A}$ 和 $\overline{B}$。

(4) 若 $|\overline{A}-\overline{B}| \leqslant T$，认为无边缘通过，进行(4.2-17)式给出的 9 点平均；若 $|\overline{A}-\overline{B}| > T$，则认为有边缘通过，$P$ 与 B 组中的 5 点进行 6 点平均。

若记 9 点均值为 $\overline{N}$，3 点均值为 $\overline{M}_3$，6 点均值为 $\overline{M}_6$，则找边的位置即找 $|\overline{M}_6-\overline{M}_3|$ 最大的情况。半邻域法的数学表达式为：

$$
g(m, n)=\begin{cases}\overline{N}, & |\overline{M}_6-\overline{M}_3|_{\max} \leqslant T \\ \overline{M}_6, & |\overline{M}_6-\overline{M}_3|_{\max} > T\end{cases} \tag{4.2-19}
$$

4. 用邻域均值与标准差定邻域的平均法

P 周围有 8 个邻点 $A_i(i=0, 1, \cdots, 7)$。对这 8 点进行统计，求出均值

$$
\mu=\frac{1}{8} \sum_{i=0}^{7} A_i \tag{4.2-20}
$$

标准偏差

$$
\sigma=\sqrt{\frac{1}{8} \sum_{i=0}^{7}(A_i-\mu)^2} \tag{4.2-21}
$$

具体算法为：

(1) 人为地设定常数因子 α。通常当 $\alpha=1$ 或 1/2 时，效果较好。

(2) 若 $\alpha|P-\mu|<\sigma$，即 σ 较大，说明 A_i 各点偏离大，可能有边缘，并且 P 与平均值 μ 接近，作出判决：取 A_i 中灰度值最接近 P 的 4 个像素进行 5 点平均。

(3) 若 $\alpha|P-\mu|>\sigma$，即 σ 较小，说明 A_i 各点灰度值偏离小，可能没有边缘在 3×3 局部区域内通过，并且 P 与平均值 μ 相差大，P 有可能为噪声点，判决取该点 8 个邻点灰度值的平均值作为该点的灰度值。

这种算法的数学表达式为：

$$g(m,n)=\begin{cases}\dfrac{1}{5}\sum\limits_{(i,j)\in S_1}f(i,j), & \alpha|P-\mu|\leqslant\sigma\\ \mu, & \alpha|P-\mu|>\sigma\end{cases} \tag{4.2-22}$$

式中：S_1 是指在 8 个邻点中挑出的与 P 值最接近的 4 个 A_i 及 P 点本身。

这种算法对冲激噪声点去除得比较快，并且可以同时保持边缘。

5. 加权平均法

如果处理窗内的各点权值不是固定的，而是按灰度值的特殊程度来确定，此时模板内的系数不同，这就是加权平均法。

(1) 邻域加权法。

P 周围有 8 个邻点 $A_i(i=0,1,\cdots,7)$。这 9 个像素灰度值加权值的确定方法为：若 $|P-A_i|\leqslant5$，加权值为 5 或 4；若 $|P-A_i|>5$，加权值为 1。然后对模板内各系数进行归一化处理，使各系数和为 1。

(2) 差敏加权法。

这个方法的基本思想是：周围像素的灰度值与 P 的灰度值差别越大，该像素灰度值的加权值就越小。例如，若灰度差值以 $|\Delta|$ 表示，则可以用 $e^{-|\Delta|}$ 的形式表示加权值。这样做大体上可以兼顾消除噪声和保护边缘两方面的要求。

下面介绍一种根据这一思想设计的算法。

在一个 3×3 的局部区域上，有下面 9 个像素：

$$\begin{matrix}a & b & c\\ d & e & f\\ g & h & i\end{matrix}$$

总的处理用模板 $[\boldsymbol{M}]$，由下述 4 个分模板构成：

$$[\boldsymbol{M}_0]=\begin{bmatrix}\alpha & \alpha & \alpha\\ 1 & 1 & 1\\ \beta & \beta & \beta\end{bmatrix}\quad [\boldsymbol{M}_{45}]=\begin{bmatrix}\gamma & \gamma & 1\\ \gamma & 1 & \delta\\ 1 & \delta & \delta\end{bmatrix}$$

$$[\boldsymbol{M}_{90}]=\begin{bmatrix}\varepsilon & 1 & \xi\\ \varepsilon & 1 & \xi\\ \varepsilon & 1 & \xi\end{bmatrix}\quad [\boldsymbol{M}_{135}]=\begin{bmatrix}1 & \eta & \eta\\ \theta & 1 & \eta\\ \theta & \theta & 1\end{bmatrix}\tag{4.2-23}$$

它们分别可以检测四个不同方向上灰度值的变化。上述分模板中的各系数值为：

$$\alpha=\exp\left(-\frac{1}{\sigma}\mid(a+b+c)-(d+e+f)\mid\right)$$

$$\beta=\exp\left(-\frac{1}{\sigma}\mid(g+h+i)-(d+e+f)\mid\right)$$

$$\gamma=\exp\left(-\frac{1}{\sigma}\mid(a+b+d)-(c+e+g)\mid\right)$$

$$\delta=\exp\left(-\frac{1}{\sigma}\mid(h+f+i)-(c+e+g)\mid\right)$$

$$\varepsilon=\exp\left(-\frac{1}{\sigma}\mid(a+d+g)-(b+e+h)\mid\right)$$

$$\xi=\exp\left(-\frac{1}{\sigma}\mid(c+f+i)-(b+e+h)\mid\right)$$

$$\eta=\exp\left(-\frac{1}{\sigma}\mid(b+c+f)-(a+e+i)\mid\right)$$

$$\theta=\exp\left(-\frac{1}{\sigma}\mid(d+h+g)-(a+e+i)\mid\right)\tag{4.2-24}$$

式中：σ 是常数，可取 3 至 6。σ 越小，平均作用越小，它由操作人员来定。

总的模板[$\boldsymbol{M}$]由四个分模板对应系数的积组成：

$$[\boldsymbol{M}']=\begin{bmatrix}\alpha\gamma\varepsilon & \alpha\gamma\eta & \alpha\xi\eta\\ \gamma\varepsilon\theta & 1 & \xi\delta\eta\\ \beta\varepsilon\theta & \beta\delta\theta & \beta\delta\xi\end{bmatrix}=\begin{bmatrix}m'_{11} & m'_{12} & m'_{13}\\ m'_{21} & m'_{22} & m'_{23}\\ m'_{31} & m'_{32} & m'_{33}\end{bmatrix}\tag{4.2-25}$$

进行归一化的方式有三种：

① 如式(4.2-23)中所示，取 $m'_{22}=1$，并按下式归一：

$$m_{ij}=\frac{m'_{ij}}{\sum_{i=1}^{3}\sum_{j=1}^{3}m'_{ij}}\tag{4.2-26}$$

② 取 $m'_{22}=\max\limits_{(i,j)\neq(2,2)}(m'_{ij})$，再按式(4.2-24)归一。

③ 归一化后，取中间系数 $m'_{22}=\frac{1}{9}$，其余 8 个系数之和为$\frac{8}{9}$，这 8 个系数保持原来的比例关系。

上述几种方法都是以邻域加权平均的设想为基础的，都可以进行迭代运算，从

而弥补 3 × 3 区域过小带来的缺点。迭代次数目前只能由实验结果判断。迭代会导致图像各点间的空间相关。通常为了节省时间，迭代次数不超过 5 次。

4.3　频域滤波

上一节从空间域的角度讨论了图像的锐化和平滑，本节从频率域的角度来讨论图像的锐化和平滑。

4.3.1　锐化模板与高频提升滤波器

图 4.21 所示的模板算法，实际上也是一个二维卷积过程。这里已表明处理系统是线性移不变的。原始二维信号为 $f(m, n)$，输出信号为 $g(m, n)$，模板则确定了一个二维滤波器。当模板为：

$$[\boldsymbol{M}] = \begin{bmatrix} m_{11} & m_{12} & m_{13} \\ m_{21} & m_{22} & m_{23} \\ m_{31} & m_{32} & m_{33} \end{bmatrix} \tag{4.3-1}$$

则对应的二维滤波器的空间域表达式为：

$$[\boldsymbol{h}(m, n)] = \begin{bmatrix} m_{33} & m_{32} & m_{31} \\ m_{23} & m_{22} & m_{21} \\ m_{13} & m_{12} & m_{11} \end{bmatrix} \tag{4.3-2}$$

并可由上式进一步求出转移函数 $H(\omega_m, \omega_n)$。下面求出由式(4.2-8)确定的图像锐化滤波器对应的转移函数。

由拉普拉斯算子确定的图像锐化算法

$$g(m, n) = (1+4\alpha)f(m, n) - \alpha[f(m-1, n) + f(m+1, n) + f(m, n-1) + f(m, n+1)]$$

可立即写出对应的 z 变换式：

$$G(z_m, z_n) = [(1+4\alpha) - \alpha(z_m^{-1} + z_m + z_n^{-1} + z_n)]F(z_m, z_n) \tag{4.3-3}$$

$$H(z_m, z_n) = \frac{G(z_m, z_n)}{F(z_m, z_n)} = (1+4\alpha) - \alpha(z_m^{-1} + z_m + z_n^{-1} + z_n) \tag{4.3-4}$$

以 $z_m = e^{j\omega_m}$，$z_n = e^{j\omega_n}$ 代入上式，得到傅里叶变换式：

$$H(\omega_m, \omega_n) = 1 + 2\alpha(1-\cos\omega_m) + 2\alpha(1-\cos\omega_n) \tag{4.3-5}$$

根据式(4.3-5)作的图,如图4.24所示。

从式(4.3-5)可知,当$\omega_m=\omega_n=0$时,$|H|$具有最小值1,这说明“直流”分量,即图像的平均灰度值处理前后不变。当$\omega_m=\omega_n=\pi$时,$|H|$有最大值$1+8\alpha$,即最高空间频率得到了最大的放大。对于M行N列数字图像$f(m, n)$,在频率域上,

$$u=\frac{M}{2}, v=\frac{N}{2}$$

对应着最高空间频率,相当于(4.3-5)式中$\omega_m=\omega_n=\pi$。

从图4.24(b)可知,该滤波器在$|H|=1$以下部分用于保留原图信息,$|H|=1$以上部分是一个二维高通滤波器。处理后的图像$g(m, n)$相当于在原图上叠加一个经高通滤波器处理后的图像。该滤波器的作用是提升图像细节,并抑制大面积上的灰度缓变成分。

式(4.3-5)是一个大于零的实函数,因此该滤波器不引入相移,即它不移动原图中目标的位置。

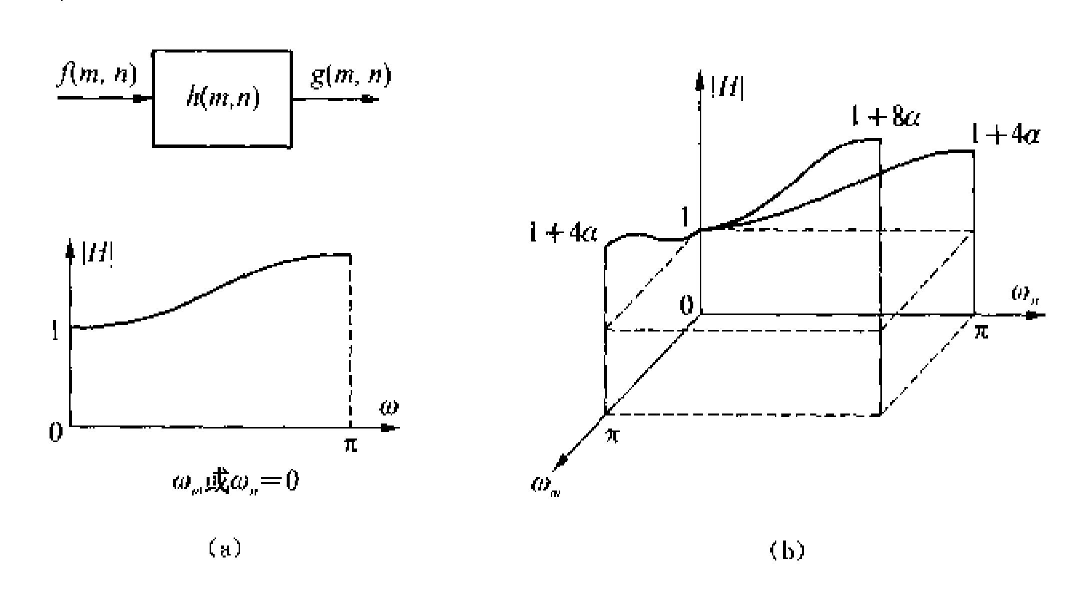


图4.24 由式(4.3-5)确定的图像锐化方法对应的滤波器的转移函数$H(u, v)$

以上讨论表明,图像的轮廓是图像中灰度值的突变部分,包含有丰富的空间高频分量。采用高频提升或高通滤波器使图像中高频分量相对提升,低频分量相对抑制,可使轮廓清晰。理想的高通滤波器如图4.25(a)实线所示。其中$D(u, v)=\sqrt{u^2+v^2}$,D_0为截止频率。由于(a)中的转移函数具有陡峭的截止特性,它会使$g(m, n)$在轮廓两侧像素的灰度值发生“抖动”现象,故实际的转移函数径向截面如(a)中虚线所示,要求平滑过渡。下面是几种典型的高通滤波器:

(1) 巴特沃思滤波器:

$$H(u, v) = \frac{1}{1+(\sqrt{2}-1)\left[\frac{D_0}{D(u, v)}\right]^{2n}} \tag{4.3-6}$$

式中：n 是滤波器的阶次；D_0 为截止频率。

（2）指数滤波器：

$$H(u, v) = \exp\left\{\left[\ln\left(\frac{1}{\sqrt{2}}\right)\right]\left[\frac{D_0}{D(u, v)}\right]^n\right\} \tag{4.3-7}$$

式中：n 控制着 $H(u, v)$的增长速率；D_0 为截止频率。

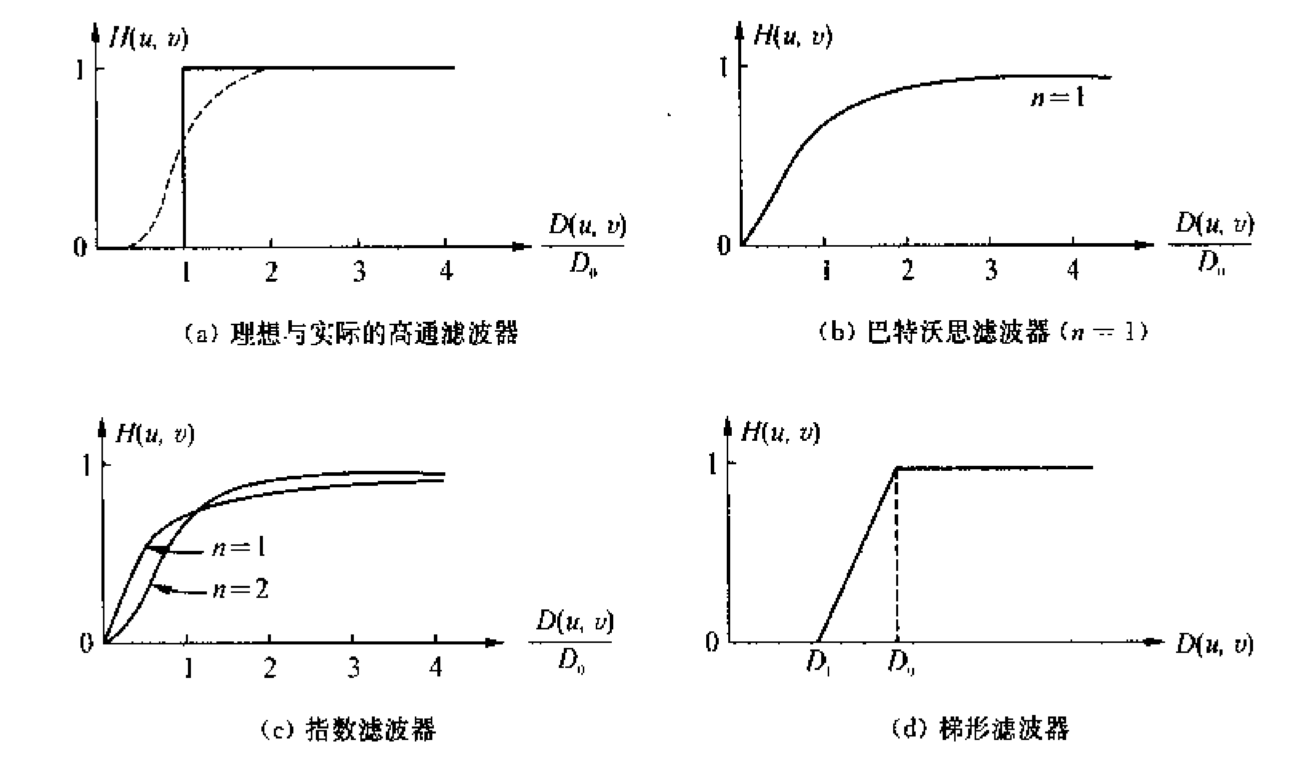


图 4.25 几种滤波器

（3）梯形滤波器：

$$H(u, v) = \begin{cases} 0, & D(u, v) < D_1 \\ \dfrac{D(u, v) - D_1}{D_0 - D_1}, & D_1 \leqslant D(u, v) \leqslant D_0 \\ 1, & D(u, v) > D_0 \end{cases} \tag{4.3-8}$$

上述三种滤波器转移函数的径向剖面示于图 4.25(b)、(c)、(d)。其中(1)、(2)中的特定常数是为了保证在空间截止频率 D_0 处，$H(u, v) = \frac{1}{\sqrt{2}}$；(3)中的 D_0、D_1 含义见图 4.25(d)。

4.3.2 平滑模板与低通滤波器

与式(4.2-8)一样，由式(4.2-13)和式(4.2-16)或式(4.2-17)确定的图像平滑滤波器也是一种线性滤波器，也可求出它们所确定的滤波器的转移函数。

由式(4.2-13)和(4.2-17)确定的图像平滑公式如下：

$$g(m,n)=\frac{1}{9}\sum_{k=-1}^{1}\sum_{l=-1}^{1}f(m-k,n-l) \tag{4.3-9}$$

可求出对应的 z 变换式为：

$$\begin{aligned}G(z_m,z_n)&=\frac{1}{9}\sum_{k=-1}^{1}\sum_{l=-1}^{1}F(z_m,z_n)z_m^k z_m^l\\&=\frac{1}{9}F(z_m,z_n)\Big(\sum_{k=-1}^{1}z_m^k\Big)\Big(\sum_{l=-1}^{1}z_n^l\Big)\end{aligned} \tag{4.3-10}$$

可得到：

$$H(z_m,z_n)=\frac{G(z_m,z_n)}{F(z_m,z_n)}=\frac{1}{9}(1+z_m+z_m^{-1})(1+z_n+z_n^{-1}) \tag{4.3-11}$$

以 $z_m=e^{j\omega_m}$，$z_n=e^{j\omega_n}$ 代入上式，得到傅里叶变换式为：

$$H(\omega_m,\omega_n)=\frac{1}{9}(1+2\cos\omega_m)(1+2\cos\omega_n) \tag{4.3-12}$$

根据式(4.3-12)作的图如图 4.26(a)和(b)所示。

由式(4.3-12)可知，当 $\omega_m=\omega_n=0$ 时，H 具有最大值 1，即处理前后图像的平均灰度值不变；当 ω_m 或 $\omega_n=\frac{2}{3}\pi$ 时，$H=0$，在这些空间频率处的信号衰减最大。从图 4.26 可以看到，在 $\omega_m\omega_n$ 平面上划分了四个区域：

Ⅰ区 $\begin{cases}0\leqslant\omega_m\leqslant\frac{2}{3}\pi\\0\leqslant\omega_n\leqslant\frac{2}{3}\pi\end{cases}$ Ⅱ区 $\begin{cases}\frac{2}{3}\pi<\omega_m\leqslant\pi\\0\leqslant\omega_n\leqslant\frac{2}{3}\pi\end{cases}$

Ⅲ区 $\begin{cases}0\leqslant\omega_m\leqslant\frac{2}{3}\pi\\\frac{2}{3}\pi<\omega_n\leqslant\pi\end{cases}$ Ⅳ区 $\begin{cases}\frac{2}{3}\pi<\omega_m\leqslant\pi\\\frac{2}{3}\pi<\omega_n\leqslant\pi\end{cases}$

Ⅱ，Ⅲ，Ⅳ区对应较高的空间频率，$|H|$ 值较小，使相应信号有较大的衰减，因此，由此确定的线性滤波器是一个低通滤波器。由于 $H(\omega_m,\omega_n)$ 是实数，在Ⅰ，

Ⅳ两区内 $H>0$，对相应的这些空间频率的信号没有相移，处理前后较大目标的位置不会有移动。但是在Ⅱ，Ⅲ两区内 $H<0$，该滤波器对相应空间频率的信号有"倒相"作用。图 4.26(b)中，B、C、D 点处的空间频率处理前后空间域中图像的变化如(c)、(d)、(e)所示。在(d)中发生了"倒相"现象，这一效应值得注意。

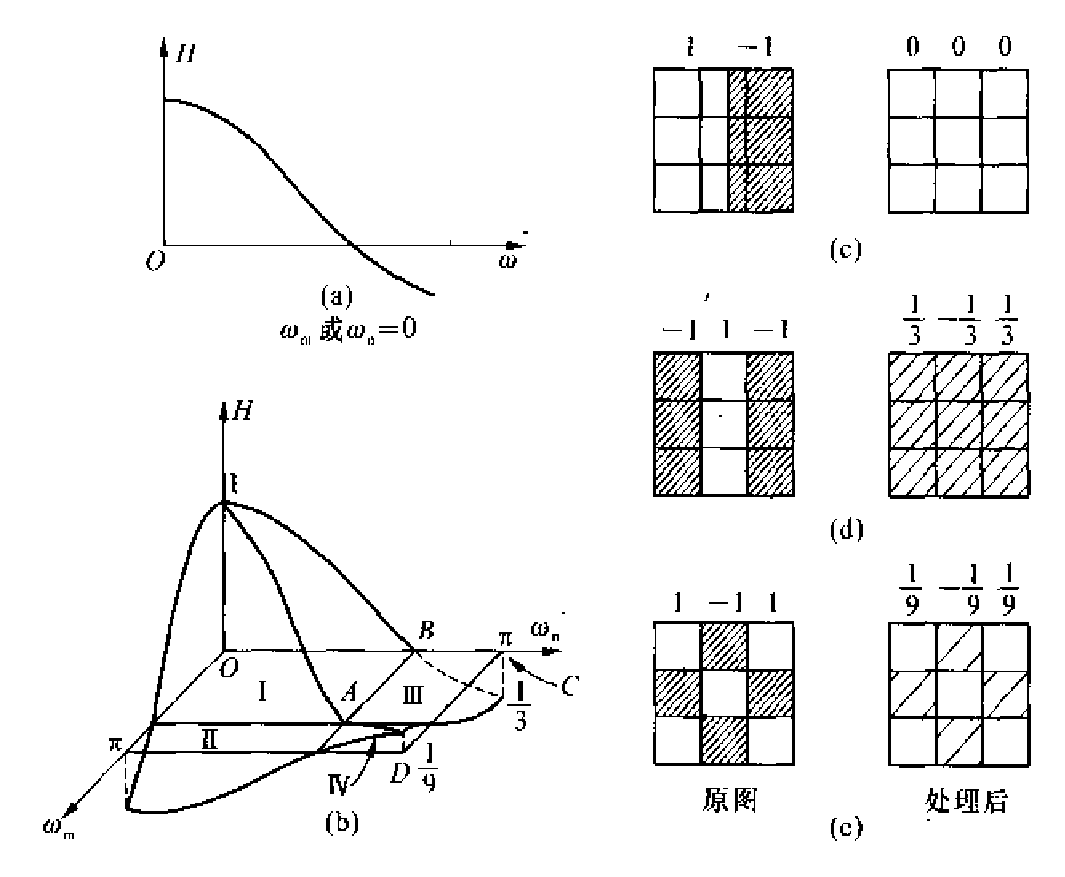


图 4.26　图像平滑算法低通滤波器

理想的低通滤波器如图 4.27(a)实线所示。其中 $D(u, v)=\sqrt{u^2+v^2}$，D_0 为截止频率。与 4.3.1 小节讨论中作的说明一样，由于图 4.27(a)中的转移函数具有陡峭的截止特性，它会使 $g(m, n)$中图像轮廓两侧像素的灰度值发生"抖动"现象，因而实际的转移函数径向截面应如图 4.27(a)中虚线所示。下面是几种典型的低通滤波器：

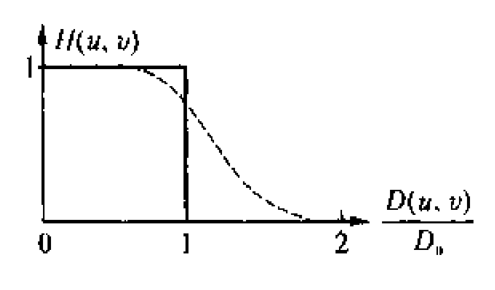
(1) 巴特沃思滤波器：

$$H(u, v)=\frac{1}{1+(\sqrt{2}-1)\left[\frac{D(u, v)}{D_0}\right]^{2n}} \tag{4.3-13}$$

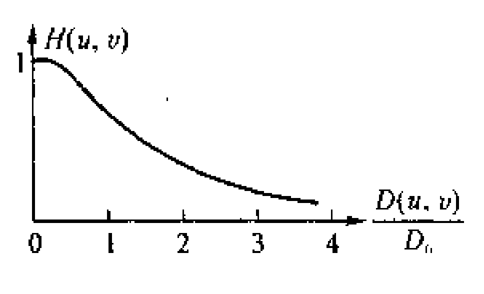
式中：n 是滤波器的阶次；D_0 为截止频率。

(2) 指数滤波器：

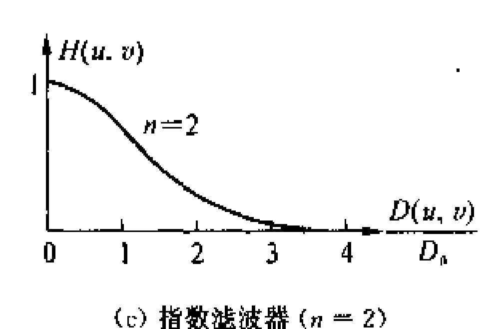
$$H(u, v)=\exp\left\{\left[\ln\left(\frac{1}{\sqrt{2}}\right)\right]\left[\frac{D(u, v)}{D_0}\right]^{n}\right\} \tag{4.3-14}$$



(a) 理想与实际的低通滤波器



(b) 巴特沃思滤波器 ($n=1$)



(c) 指数滤波器 ($n=2$)

H(u, v)
1
0
D_0
D_1
D(u, v)

(d) 梯形滤波器

图 4.27 几种低通滤波器

式中：n 为阶数；D_0 为截止频率。n 控制着 $H(u, v)$ 的衰减速率。

(3) 梯形滤波器：

$$H(u, v)=\begin{cases} 0, & D(u, v)<D_0 \\ \dfrac{D(u, v)-D_1}{D_0-D_1}, & D_0 \leqslant D(u, v) \leqslant D_1 \\ 1, & D(u, v)>D_1 \end{cases} \quad (4.3-15)$$

式中：$D_0<D_1$，D_0 为截止频率。

4.4 中值滤波

中值滤波(Median Filtering)，有的书上也称中值平滑(Median Smoother)，是一种空间域非线性滤波技术。

4.4.1 用于平滑噪声的中值滤波

1. 中值滤波的概念

中值滤波也可用于图像平滑，它能够在滤除噪声的同时保持边缘不被模糊。

中值滤波的做法是：以处理窗内原图灰度值的"中值"作为"窗口"中心处的新值。

以一维5点中值滤波为例：若原信号序列为"80，90，200，110，120"，即斜坡变化的中心处有大孤立噪声。做中值滤波时，取5个数的中值（即第3大的值）"110"作为滤波后原图"200"所在的中心位置处的新值。

由上例可见，不管孤立噪声有多大，都可以被彻底去除，而不像作邻域平均平滑时还保留了噪声值的$1/M$。

若原序列为"100，100，200，200，200"，则中值滤波后中心位置处仍为"200"，不像邻域平均时会使边缘变模糊。若噪声不是孤立的，而是连续两个相等的值出现，如"100，200，200，100，100"，则仍能用中值滤波完全滤除。但若5点中有3个取值相同的噪声点连在一起，用中值滤波就无法去除了。总之，若像素变化的空间尺寸不大于（窗宽－1）/2，则可用中值滤波完全滤去。如果这种像素变化不是噪声，而是原图的细节，同样也会被滤去，这是不希望出现的。因此，应精心选择处理窗的大小，尽量在去除噪声的同时又能保留图像中的细节。显然，如果细节的尺寸小于噪声的尺寸，处理起来会很困难。

从中值滤波的原理可见，中值滤波也是一种顺序滤波器。

2. 二维中值滤波

在二维情况下，中值滤波是以二维窗内像素灰度值的中值作为窗中心处的新值，即

$$g(m,n)=\underset{i,j\in S}{\mathrm{Median}}[f(i,j)],\ S\text{为窗，中心坐标为}(m,n) \quad (4.4-1)$$

或

$$g(m,n)=\underset{i,j\in S}{\mathrm{Median}}[f(m-i,n-j)],\ S\text{为窗，中心坐标为}(0,0) \quad (4.4-2)$$

常用的二维窗形状有十字窗、×形窗和正方窗，如图4.28所示。

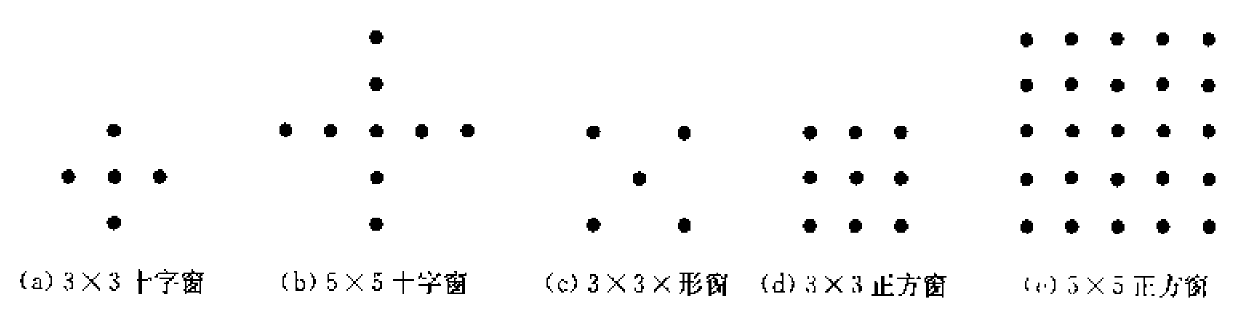


图4.28 二维中值滤波常用的处理窗

其中，用正方窗进行中值滤波时，也能在去除噪声的同时保持边缘，但常会出

现“缺角”的现象，如图 4.29(b)所示。但正方窗有利于去除颗粒噪声。而十字窗对 0 度、90 度线和角点有保护作用，如图 4.29(c)所示。×形窗则对 45°线、135°线和角点有保护作用。

(a) 原图					(b) 用 3×3 正方窗处理后的图像					(c) 用 3×3 十字窗处理后的图像				
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
1	5	5	5	1	1	1	5	1	1	1	5	5	5	1
1	5	5	5	1	1	5	5	5	1	1	5	5	5	1
1	5	5	5	1	1	1	5	1	1	1	5	5	5	1
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

图 4.29 用正方窗和十字窗进行中值滤波举例

为了能够在进行中值滤波时自适应地选用合适的窗的形状，有人提出了多窗口中值滤波法——MWMF(Multi-Windows Median Filter)算法。

对于中心点为 O 的处理窗：

$$\begin{bmatrix} a & b & c \\ h & O & d \\ g & f & e \end{bmatrix}$$

MWMF 算法的处理步骤如下：

(1) 计算以 $O(m, n)$ 为中心的 3×3 十字窗内的像素中值 md_1：

$$md_1 = \text{Median}\{b, d, f, h, O\} \tag{4.4-3}$$

(2) 计算以 $O(m, n)$ 为中心的 3×3“×”形窗内的像素中值 md_2：

$$md_2 = \text{Median}\{a, c, e, g, O\} \tag{4.4-4}$$

(3) 计算 9 点平均：

$$av = \frac{1}{9}[a + b + \cdots + h + O] \tag{4.4-5}$$

(4) 计算三个差值：

$$\begin{aligned} \Delta_1 &= md_1 - O(m, n) \\ \Delta_2 &= md_2 - O(m, n) \\ \Delta_3 &= O(m, n) - av \end{aligned} \tag{4.4-6}$$

(5) 窗口中心处的新值为：

$$g(m, n) = \begin{cases} md_1, & \text{若 } \Delta_1 = \Delta_2 \\ \max(md_1, md_2), & \text{若 } \Delta_1 \neq \Delta_2, \Delta_3 \geqslant 0 \\ \min(md_1, md_2), & \text{若 } \Delta_1 \neq \Delta_2, \Delta_3 < 0 \end{cases} \tag{4.4-7}$$

需要说明的是：二维中值滤波是不可分的，即先在 x 方向做一次中值滤波，再在 y 方向做一次中值滤波，所得结果与直接做 3×3 方窗的中值滤波的结果不尽相同。对序列图像的静止区可进行时间上的中值滤波，例如从序列图像 $f_{t_i}(m, n)$中取 $f_{-1}(m, n)$、$f_0(m, n)$、$f_1(m, n)$的中值作为结果。

4.4.2 广义的中值滤波[3]

中值滤波是一种非线性滤波器。传统的中值滤波在处理窗中取中值做平滑的做法类似于线性滤波中的"简单平均平滑"。二者都是使准则函数

$$G(\beta) = \sum_{i=1}^{N} |x_i - \beta|^p \tag{4.4-8}$$

为最小。(4.4-8)式中 $x_i(i = 1 \sim N)$ 是窗内像素值，β是要求的滤波后的中心点像素值，它使$G(\beta)$为最小。$G(\beta)$形如敏可夫斯基距离，指β点与$\{x_i\}$点集在某种意义下的距离。若 $p = 2$，即为欧氏距离，取β为"重心"可使欧氏距离为最小，此即简单平均做法的依据；若 $p = 1$，为使 $G(\beta)$为最小就应取中值。

若进一步从统计意义上看问题，做参数估计时，若随机变量为高斯分布，它的最大似然估计为均值；若随机变量为拉普拉斯分布，则其最大似然估计为中值，所以对概率分布拖尾大的噪声用中值滤波来滤除是合乎想象的。

以上的讨论只是通过中值/均值的类比来开拓思路，下面则要通过线性滤波中的一些做法上的扩展通过类比引申到"中值"情况来引出新的做法。

1. 加权中值平滑

这里的加权中值平滑(Weighted Median Smoother, WMS)只讨论加权系数为正整数的情况。

在线性滤波中，从$\frac{1}{5}\begin{bmatrix}0 & 1 & 0\\1 & 1 & 1\\0 & 1 & 0\end{bmatrix}$可引伸到$\frac{1}{3}\begin{bmatrix}0 & 0.5 & 0\\0.5 & 1 & 0.5\\0 & 0.5 & 0\end{bmatrix}$，后者也是一种平滑，但加权系数使平滑程度小于前者的简单平均。同样，在中值平滑中也可借用"加权中值"的做法来增加其灵活性，具体描述如下。

设加权窗内共有 N 个像素，可选定一个 N 维向量

$$\boldsymbol{w} = \begin{bmatrix}w_1\\w_2\\\vdots\\w_N\end{bmatrix}$$

其中加权系数 w_i 暂时限定为正整数。不同位置的样本应合理选择不同的权值。例如:靠近中心处权值最大;垂直、水平位置处权值中等;对角点权值最小。

可定义加权中值平滑为:

当 w_i 为整数时,则

$$y(n) = \text{Median}[w_1 \Diamond x_1(n),\ w_2 \Diamond x_2(n),\ \cdots,\ w_N \Diamond x_N(n)] \quad (4.4-9)$$

式(4.4-9)中 $y(n)$是经加权中值平滑后中心点 n 位置上的值;Median[·]是"取中值运算"符;$x_i(n)$是以 n 为中心的窗内第 i 个像素值;$w_i \Diamond$表示重复 w_i 次,如 $3\Diamond x_i$即 x_i, x_i, x_i,也就是把一个像素值当成 3 个来计算中值。重复计算的思想是多次重复重要的点,如对中心附近的点 w_i 取大些,多重复几次,使中心附近的点易被取为中值。

例 1 某个加权中值平滑的 $N=5$,即一维窗内有 5 个点,取权向量 $\boldsymbol{w}=[1, 2, 3, 2, 1]^T$,即认为越靠近中心的点越重要。对某个"窗"内观察向量 $\boldsymbol{x}=[12, 6, 4, 1, 9]^T$,求其经加权中值平滑后的结果。

解:如果不进行加权,则中值为 6。而进行加权中值平滑的结果为:

$$\begin{aligned} y(n) &= \text{Median}[1\Diamond 12,\ 2\Diamond 6,\ 3\Diamond 4,\ 2\Diamond 1,\ 1\Diamond 9] \\ &= \text{Median}[12;\ 6,\ 6;\ 4,\ 4,\ 4;\ 1,\ 1;\ 9] \\ &= \text{Median}[1,\ 1,\ 4,\ 4,\ 4,\ 6,\ 6,\ 9,\ 12] \\ &= 4 \end{aligned}$$

由于有可能出现由偶数个值来求中值的情况,故定义中值为:

先对窗内的值进行排序,若记入重复后共有 N 个数,排序成 $x_{(1)}$, $x_{(2)}$, …, $x_{(N)}$,则中值

$$\text{Median}[\cdot] = \begin{cases} x_{(N+1)/2}, & \text{若 } N \text{ 为奇数} \\ \dfrac{[x_{(N/2)} + x_{(N/2+1)}]}{2}, & \text{若 } N \text{ 为偶数} \end{cases} \quad (4.4-10)$$

例如当 $N=4$ 时,"4, 6, 8, 10"的中值为 $\dfrac{6+8}{2}=7$。

若用加权中值,在二维中值滤波时可保留角点,对图 4.31(a),若取

$$[\boldsymbol{W}_{ij}] = \begin{bmatrix} 1 & 1 & 1 \\ 1 & 3 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

则图 4.31(a)中的边和角点都可被保留。

2. 当权系数为正实数时的加权中值平滑

当 w_i 不是整数时,若通过简单地放大 w_i,使其系数全为整数,从原理上讲是

可以的，所有的 w_i 成比例的倍增不会改变中值。例如，“1，2，3”的中值为 2；“1，1，2，2，3，3”的中值为 $\frac{2+2}{2}$，仍为 2。但排序速度会变慢很多，所以要改变做法。

在权系数为正实数时，求中值的算法步骤为：

(1) 计算门限 $T_0 = \frac{1}{2}\sum_{i=1}^{N} w_i$，即求权值总和的 1/2 作为预设门限。

(2) 对窗内观察样本值进行从小到大的排序。

(3) 从样本(像素)值最大处开始，累加权值，得到权值的部分和 sum。

(4) 若权的部分和 sum $\geqslant T_0$，则此时最后加的“权”所对应的“像素值”即为所求的“加权中值”，否则重复(3)。

例 2　已知 $x(n)=[12, 6, 4, 1, 9]$，$w=[0.1, 0.1, 0.2, 0.2, 0.1]$，求加权中值。

解：计算门限：$T_0 = \frac{1}{2}\sum_{i=1}^{N} w_i = \frac{1}{2}(0.1+0.1+0.2+0.2+0.1) = 0.35$

排序后样本为：　　　“1，4，6，9，12”

对应的权值为：　　　“0.2，0.2，0.1，0.1，0.1”

权的部分和 sum 的值依次为：　　“0.1，0.2，0.3，0.5，0.7”

由于与像素值“4”对应的权值“0.2”加上去后，sum $= 0.5 \geqslant T_0 = 0.35$，所以求得加权中值为 4。

3. 递归中值滤波和递归加权中值滤波

递归中值滤波(Recursive Median Smoother/Filter)是指在完成了部分像素的中值滤波后，窗移一位做下一个中值滤波时，将已经过滤波的样本值作为以后滤波器的输入信号，即成“递归”的。即下次求中值时用到上次的结果。

若用式子来进行表示，则可描述如下。

当前点处于 n 位置的原图像观察向量为：

$$\boldsymbol{x}(n) = [x(n-N_L), \cdots, x(n), \cdots, x(n+N_R)] \qquad (4.4-11)$$

式中：N_L、N_R 为非负整数，总长度 $N = N_L + N_R + 1$。

普通(非递归)中值滤波输出中心点 n 位置的值为：

$$y(n) = \mathrm{Median}[x(n-N_L), \cdots, x(n), \cdots, x(n+N_R)] \qquad (4.4-12)$$

递归中值滤波的输出为：

$$y(n) = \mathrm{Median}[y(n-N_L), \cdots, y(n-1), x(n), \cdots, x(n+N_R)] \qquad (4.4-13)$$

式中：$y(n-N_L), \cdots, y(n-1)$ 为上几次滤波结果；$x(n), \cdots, x(n+N_R)$ 为原始输入。

对递归加权中值滤波，若记中心点为 $x(n)$，最左为 $x(n-N_1)$，最右为 $x(n+N_1)$，则输入为：

$$x(n-N_1),\ \cdots,\ x(n-1),\ x(n),\ \cdots,\ x(n+N_1)$$

对应的权为：

$$w_{-N_1},\ \cdots,\ w_{-1},\ w_0,\ w_1,\ \cdots,\ w_{N_1}$$

则递归加权中值滤波的输出为：

$$\begin{aligned} y(n) = \text{Median}[&w_{-N_1}\Diamond y(n-N_1),\ \cdots,\ w_{-1}\Diamond y(n-1), \\ &w_0\Diamond x(n),\ \cdots,\ w_{N_1}\Diamond x(n+N_1)] \end{aligned} \tag{4.4-14}$$

从上述分析可以得出结论：为达到同样的滤波效果，递归中值滤波器的长度比不递归的短，但可达到同样的平滑噪声作用。

4. 中心加权中值平滑

中心加权平滑(Center Weighted Median Smoother)是加权中值平滑的一个特例，即只对中心位置处加权。中心位置用 c 下标来表示，$c=\dfrac{N+1}{2}$，即上例中的 N_1+1。则中心加权平滑的输出为：

$$y(n) = \text{Median}[x_1,\ \cdots,\ x_{c-1},\ w_c\Diamond x_c,\ x_{c+1}\cdots,\ x_N] \tag{4.4-15}$$

若 $w_c=1$，则退化为一般中值滤波；若 $w_c\geqslant N$，则 $y(n)=x(n)$，不起滤波作用。

5. 带有负权值的中值滤波器

以上所说的中值滤波，权值都是正的，与线性 FIR 滤波器中的平滑或低通相对应。它可以去噪声(平滑)，且保持边缘不被模糊，但不能增强高频分量。而在线性滤波中，要达到高通的效果就要有负的加权值。相应地，带有负权值的中值滤波器也可有高通特性，可以用于边缘增强，即图像锐化。

求平均(加权)改为求“中值”，会收到更好的效果。因为现在的做法可整可实，可正可负，使 WMF 能力大大扩展了。

有负权值的加权中值滤波器的实现可描述如下。

关键要将负权值理解成 $|w_i|\Diamond \text{sgn}(w_i)x_i$，则

$$y = \text{Median}[|w_1|\Diamond \text{sgn}(w_1)x_1,\ \cdots,\ |w_N|\Diamond \text{sgn}(w_N)x_N] \tag{4.4-16}$$

即负权值“乘”正样本值＝正权值“乘”负的样本值。这里“乘”的含义，当 $|w_i|$ 为整数时为“重复计算”；当 $|w_i|$ 为实数时，要用累加权值与门限相比较来求得中值。

例如，若 5 点像素值向量 $x=[1,\ 2,\ 3,\ 4,\ 5]^T$，权向量为 $\boldsymbol{w}=[1,\ 1,\ 1,\ -1,\ 1]$，

则[−4，1，2，3，5]的中值为“2”。

例 3　若 $\boldsymbol{w}=[0.1,0.2,0.3,-0.2,0.1]$，样本值向量 $\boldsymbol{x}=[-2,2,-1,3,6]^{\mathrm{T}}$，求中值。

解：同样要按例 2 中的 4 步计算中值，只是把“负号”从“权”改到相应的样本值上去。具体步骤如下：

计算门限：　　$T_0=\dfrac{1}{2}\sum\limits_{i=1}^{N}|w_i|=0.45$

样本值排序后为：　　“−3，−2，−1，2，6”

相应的权值为：　　“0.2，0.1，0.3，0.2，0.1”

权的部分和 sum 的值依次为：　　“0.1，0.3，0.6，…”

由于与像素值“−1”对应的权值加上去后 $\mathrm{sum}=0.6\geqslant T_0=0.45$，则所求中值为“−1”。

6. 中值滤波用于图像内插

在 2.1.2 小节讨论过用内插函数(线性滤波法)来进行图像插值，如双线性内插等。现在要从线性滤波拓展到中值滤波。

中值滤波用于图像内插的步骤如下：

(1) 先开好将原图放大一倍的存储空间来存储内插后的图像，如图 4.30 所示，图中像素分为以下 4 类：

00：已知像素值 $a_{i,j}$；

01：左右为已知像素值(线性内插时取左右平均)；

10：上下为已知像素值(线性内插时取上下平均)；

11：中心点，4 个角点为已知样本值(线性内插时作 4 点平均)。

$x_{i,j}$ 00	01	$x_{i,j+1}$ 00	01	00
10	11	10	11	10
$x_{i+1,j}$ 00	01	$x_{i+1,j+1}$ 00	01	00
10	11	10	11	10
00	01	00	01	00

图 4.30　中值滤波用于图像内插

(2) 对 00 类像素填上已知值 $x_{i,j}^{00}=a_{i,j}$。

(3) 由 4 个角点的中值来填充"中心"点的值，即

$$x_{i,j}^{11}=\text{Median}[a_{i,j},\ a_{i-1,j},\ a_{i,j-1},\ a_{i-1,j-1}] \tag{4.4-17}$$

(4) 对"01"点进行中值内插。用加权中值滤波对左右的已知点用权 1，对上下的 x^{11} 点用权值 0.5，即

$$x_{i,j}^{01}=\text{Median}[a_{i,j},\ a_{i,j+1},\ 0.5\Diamond x_{i-1,j}^{11},\ 0.5\Diamond x_{i,j}^{11}] \tag{4.4-18}$$

(5) 同理，对"10"点进行中值内插：

$$x_{i,j}^{10}=\text{Median}[a_{i,j},\ a_{i-1,j},\ 0.5\Diamond x_{i,j-1}^{11},\ 0.5\Diamond x_{i,j}^{11}] \tag{4.4-19}$$

本小节内容取材于文献[3]，有兴趣对此做深入研究的读者可参阅该文献。

4.5 图像增强的其他方法

4.5.1 同态滤波

图像 $f(x, y)$是由光源产生的照度场 $f_i(x, y)$和目标(景物或照片)的反射系数场 $f_r(x, y)$共同作用下产生的，三者的关系是：

$$f(x, y)=f_i(x, y)f_r(x, y) \tag{4.5-1}$$

理想情况下，照度场 $f_i(x, y)$是一个常数，这时 $f(x, y)$可以不失真地反映 $f_r(x, y)$。然而在一般情况下，由于光照不均匀，$f_i(x, y)$不是常数，其值随坐标的变化而缓慢地变化。此外，光传输系统和光电转换设备(如摄像机)的不完善，可以造成类似于照度场不均匀的效果，也可等效看成照度场的不均匀。这样会造成图像$f(x, y)$上有大面积阴影。同态滤波可以在保留图像细节的同时，消除这些大面积阴影，以提高图像在暗区目标的清晰度。

同态滤波是数字信号处理中的同态信号处理在图像增强中的具体应用，其原理可描述如下。

由于光源照度场 $f_i(x, y)$的变化缓慢，在频谱上其能量集中于低频；反射系数场 $f_r(x, y)$包含了所需要的图像信息，它在空间的变化较快，其能量集中于高频。因此，采用同态分析方法可把这两个分量变为相加的两个分量分开处理，以期压缩对比度，并可压缩图像信号的动态范围。处理步骤如下：

(1) 对式(4.5-1)两边取对数，获得两个加性分量。这是同态分析方法中的

关键一步：

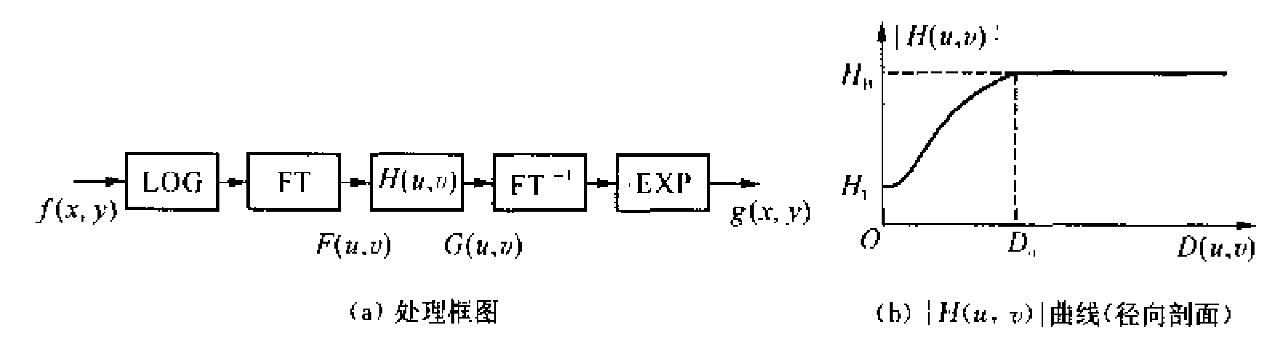
$$\ln f(x, y) = \ln f_i(x, y) + \ln f_r(x, y) \tag{4.5-2}$$

(2) 对(4.5-2)式两边进行傅里叶变换，得到

$$F(u, v) = \mathscr{F}\{\ln f_i(x, y)\} + \mathscr{F}\{\ln f_r(x, y)\} = F_i(u, v) + F_r(u, v) \tag{4.5-3}$$

在这里，$F_i(u, v)$和$F_r(u, v)$并不直接是$\mathscr{F}\{\ln f_i(x, y)\}$和$\mathscr{F}\{\ln f_r(x, y)\}$，但由于ln{·}是单调增函数，可以认为$F_i(u, v)$和$F_r(u, v)$分别在频谱的低频区和高频区的这个格局并没有很大变化。

(3) 为了消除照度的起伏，应压缩$F_i(u, v)$分量。为了增加图像本身的对比度，应增强$F_r(u, v)$分量。因此应当使用一个具有图4.31(b)那样特性的滤波器来处理$F(u, v)$。图中$D(u, v) = \sqrt{u^2 + v^2}$。



(a) 处理框图　　(b) $|H(u, v)|$曲线(径向剖面)

图4.31 同态滤波器

$$G(u, v) = H(u, v)F(u, v) = H(u, v)F_i(u, v) + H(u, v)F_r(u, v) \tag{4.5-4}$$

其中滤波器的输出是$G(u, v) = \mathscr{F}\{\ln g(x, y)\}$。

(4) 取傅里叶反变换和指数运算，得到输出图像$g(x, y)$：

$$\begin{aligned}\ln g(x, y) &= \mathscr{F}^{-1}\{G(u, v)\} \\ &= \mathscr{F}^{-1}\{H(u, v)F_i(u, v)\} + \mathscr{F}^{-1}\{H(u, v)F_r(u, v)\}\end{aligned} \tag{4.5-5}$$

$$g(x, y) = \exp[\mathscr{F}^{-1}\{HF_i\}]\exp[\mathscr{F}^{-1}\{HF_r\}] \tag{4.5-6}$$

综合式(4.5-1)至(4.5-6)可写出：

$$g(x, y) = \exp[\mathscr{F}^{-1}\{H(u, v) \cdot \mathscr{F}[\ln f(x, y)]\}] \tag{4.5-7}$$

即图4.31(a)框图的表达式。

同态处理的效果与$|H(u,v)|\sim D(u,v)$曲线的三个参数H_H、H_L和D_0的选择有关。为了压缩照度分量$F_i(u,v)$，应取$0<H_L<1$；为增强反射系数分量$F_r(u,v)$，应取$H_H>1$，例如取$H_L=0.5$，$H_H=2$。D_0的选择较困难，它与照度场$f_i(x,y)$及反射系数场$f_r(x,y)$对应的频谱幅度对比有关，一般需经多次试验才能确定。

4.5.2 小波去噪

小波去噪是图像平滑的一类方法，它是在对图像进行小波变换的基础上，在变换域中进行处理，以去除图像中的噪声，改善图像的质量。由于小波分解的多分辨率分析特点与人眼视觉系统(HVS)信息处理的多通道滤波机制相类似，小波去噪已经成为变换域图像去噪的主要研究方向之一。

在数学上，小波去噪问题的本质是一个函数逼近问题，以及如何在小波母函数伸缩和平移版本所展成的函数空间中，根据提出的衡量准则，寻找对原信号的最佳逼近，以完成原信号和噪声信号的区分。从信号学的角度看，小波去噪是一个信号滤波的问题，而且尽管在很大程度上小波去噪可以看成低通滤波，但是由于在去噪后还能成功地保留图像特征，所以在这一点上又优于传统的低通滤波器。由此可见，小波去噪实际上是特征提取和低通滤波功能的综合，其流程图如图4.32所示。

小波去噪的方法有很多种。1988年，Mallat提出了多分辨率分析的概念，使小波具有带通滤波的特性，因此可以利用小波分解与重构的方法滤波降噪；1992年，Mallat又提出了奇异性检测的理论，从而可利用小波变换模极大值的方法去噪；1995年，Donoho提出了小波变换阈值法去噪，用该方法去噪得到了非常广泛的应用；随后，在阈值法的基础上又提出了平移不变量小波去噪法。此外，将多小波(Multi-wavelet)用于去噪的方法近年来也日益成熟。目前，如何选取阈值，针对不同类型噪声的小波域图像去噪，以及小波变换与其他方法结合去噪的研究仍然十分活跃。

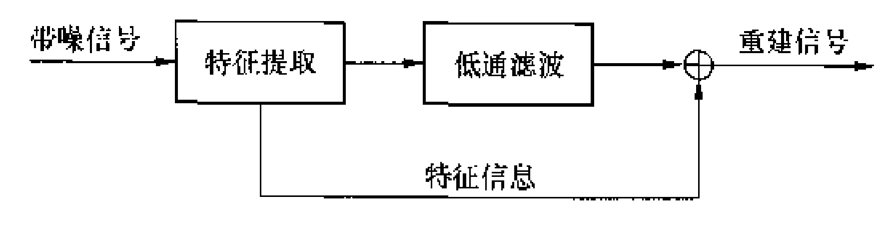


图4.32 小波去噪流程图

鉴于小波变换阈值去噪方法的简单和有效性，本节中只对小波变换阈值去噪方法的原理和特点进行具体介绍。

小波变换阈值去噪法也称为“小波收缩”(Wavelet Shrinkage)，其去噪原理可描述如下：

假设有一信号 $f(k)$ 表达式为：

$$f(k)=s(k)+n(k) \tag{4.5-8}$$

式中：$s(k)$ 为纯净信号；$n(k)$ 为加性随机噪声。

其小波变换为：

$$w(k)=\theta(k)+z(k) \tag{4.5-9}$$

式中：w，θ 和 z 分别为信号、纯净信号以及噪声的小波系数。

小波阈值去噪方法为：设置一个阈值(由先验知识得到)，大于这个阈值的小波系数认为是由信号产生的，小于这个阈值的小波系数认为是由噪声产生的，则去除那些由噪声产生的系数就可以达到去噪的目的。

小波阈值去噪法分为以下三个步骤：

(1) 计算含噪声信号的正交小波变换。选择合适的小波和小波分解层数 j，将含噪信号进行小波分解至 j 层，得到相应的小波分解系数。

(2) 对分解得到的小波系数进行阈值处理，得到纯净信号小波系数的估值。

这里进行阈值处理的阈值函数有很多种，最简单的有硬阈值函数和软阈值函数，采用硬阈值函数和软阈值函数的去噪方法分别称为硬阈值法和软阈值法。硬阈值函数和软阈值函数的定义分别如下：

$$\text{硬阈值函数：}\sigma_\lambda^H(w)=\begin{cases}w, & |w|>\lambda\\ 0, & |w|\leqslant\lambda\end{cases} \tag{4.5-10}$$

$$\text{软阈值函数：}\sigma_\lambda^S(w)=\begin{cases}\operatorname{sgn}(w)(|w|-\lambda), & |w|>\lambda\\ 0, & |w|\leqslant\lambda\end{cases} \tag{4.5-11}$$

式中：w 为小波系数；λ 为阈值；$\sigma(w)$ 为纯净信号小波系数的估计值。

(3) 进行小波逆变换。将经阈值处理过的小波系数估值进行重构，得到恢复的纯净信号估计值。

这种方法主要适用于信号中混有白噪声的情况。由于小波变换具有一种“集中”的能力，它能将信号的能量集中到少数小波系数上，而白噪声在任何正交基上的变换仍然是白噪声，并且有着相同的幅度。相对来说，信号的小波系数值必然大于那些能量分散且幅值较小的噪声的小波系数值。选择一个合适的阈值，对小波系数进行阈值处理，就可以达到去除噪声而保留有用信号的目的。该方法能得到原始信号的近似最优估计，并且具有非常广泛的适用性。

用小波阈值法去噪的优点是噪声几乎完全得到抑制，且反映原始信号的特征尖峰点得到很好的保留。Donoho 通过实验证明，用软阈值的方法去噪能够使

处理后的估计信号实现最大均方误差最小化，即去噪后的估计信号是原始信号的近似最优估计；且估计信号至少和原始信号同样光滑而不会产生附加震荡。该方法还具有广泛的适用性，因而是众多小波去噪方法中应用最为广泛的一种。小波阈值去噪法的另一个优点是计算速度很快，计算复杂度为 $O(N)$，其中 N 为信号长度。其缺点是在有些情况下，如在信号的不连续点处，去噪后会出现吉布斯(Gibbs)现象，即不连续点附近的信号会在一个特定的目标水平上下跳变，表现出视觉上的非自然信号。在用该方法去噪时，阈值的选择对去噪效果有着很重要的影响。阈值的选择方法有多种，实际应用时应根据具体的情况来选择合适的阈值。

目前使用的阈值可以分为全局阈值和局部阈值两类。其中，全局阈值对各层所有的小波系数或同一层内的小波系数都是统一的；而局部阈值则是根据当前系数周围的局部情况来确定阈值。

全局阈值主要有统一阈值、置信区间阈值、最小最大化阈值、理想阈值、BayesShrink 阈值和 MapShrink 阈值等。其中，由于统一阈值计算简单，故得到广泛的应用，但是它趋向于"过扼杀"小波系数，从而会导致较大的重建误差。置信区间阈值虽然和图像尺寸无关，但由于图像尺寸增大时，大的噪声系数出现的数目会增多，并被保留，因而导致误差增大。实验表明，BayesShrink 阈值能够获得接近于理想阈值的去噪结果。另外，人们有时往往根据待处理信号的一些先验知识，结合经验来给出不同的阈值。

与全局阈值不同，局部阈值主要是通过考查在某一点或某一局部的特点，再根据灵活的判定原则来判定系数是"主噪"还是"主信"，以实现"去噪"和"保留信号"之间的平衡。这些判定原则有时并不一定是从系数的绝对值来考虑，而是从其他方面，例如从概率和模糊隶属度方面来考虑。实验结果表明，局部阈值确实比全局阈值对信号的适应能力好，但是需要较为繁琐的计算。

目前，对阈值选择方法的研究仍在进行中，仍有新的阈值公式不断被提出。通常，阈值是根据实际应用的需要，通过确定合适的准则，并通过对可能的阈值进行寻优来选择。

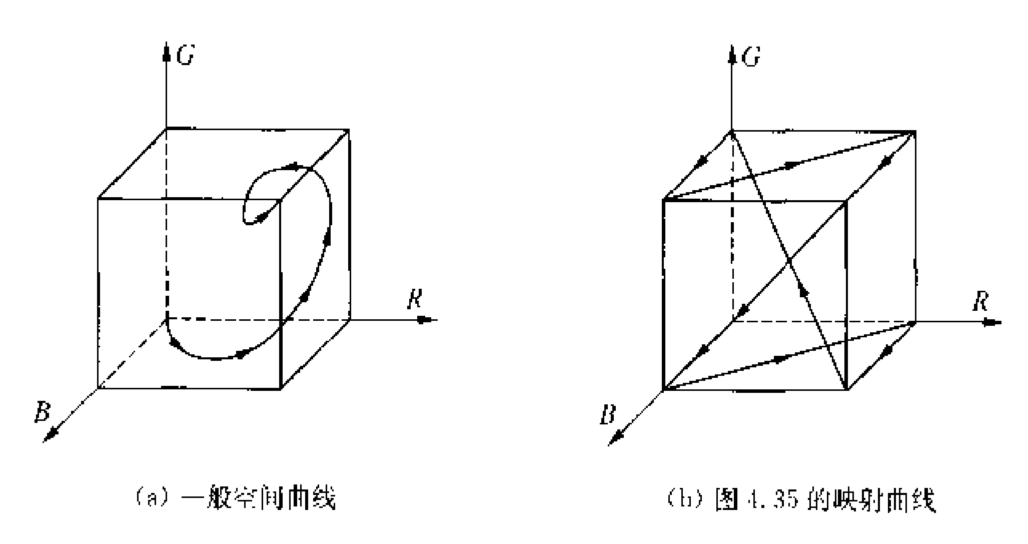
4.5.3 伪彩色增强

数字图像的一个像素通常占有 8 比特，可以有 256 级灰度层次。但是当人眼对某一背景或平均亮度适应之后，可分辨的灰度等级数还要下降。然而人眼对不同色度和亮度的彩色的分辨能力却很强。若把黑白图像的灰度值按某种函数关系映射成对应的彩色，即作伪彩色处理，则可大大提高观察者对图内目标的检测能力。

根据2.4.2小节中所述三基色原理，就人眼主观感觉而言，任何彩色可由红、绿、蓝三基色按一定比例合成。如果以$R(m, n)$，$G(m, n)$，$B(m, n)$表示黑白(单谱)图像中某一像素的灰度值$f(m, n)$映射为伪彩色时的三个基色分量的值，则伪彩色增强可定义为：

$$\begin{cases} R(m, n) = T_R[f(m, n)] \\ G(m, n) = T_G[f(m, n)] \\ B(m, n) = T_B[f(m, n)] \end{cases} \tag{4.5-12}$$

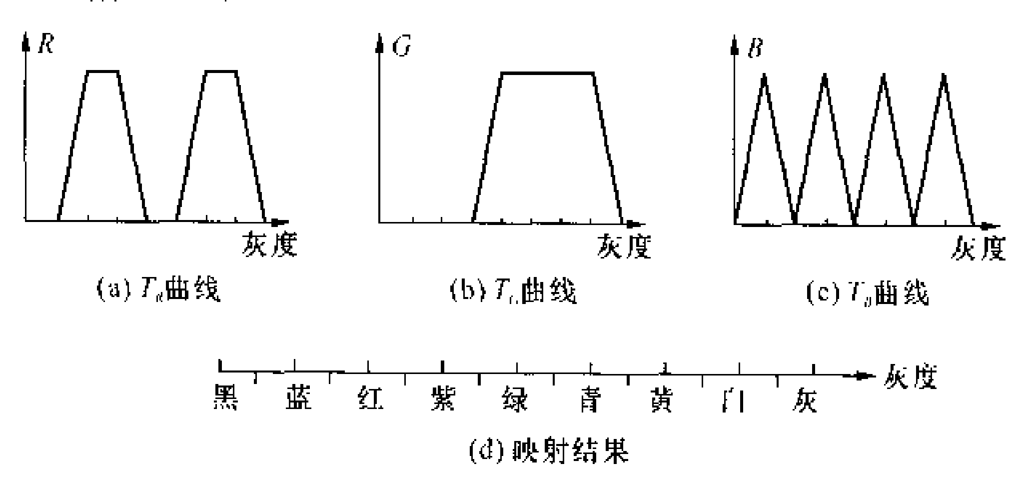
式中：T_R，T_G，T_B是原图像灰度值的线性或非线性函数。当灰度值逐步递增时，由式(4.5-12)确定了R，G，B色立体中的一条空间曲线，见图4.33。



(a) 一般空间曲线　　(b) 图4.35的映射曲线

图4.33　色立体中黑白图片的伪彩色映射

与灰度增强一样，伪彩色增强也不改变原来像素的位置，如式(4.5-12)所示，它是“一对三”的映射。图4.34中，当三个函数曲线分别为(a)，(b)，(c)时，所完成的伪彩色映射如(d)所示。



(a) T_R曲线　　(b) T_G曲线　　(c) T_B曲线

(d) 映射结果

图4.34　伪彩色映射例子

4.5.4 假彩色增强

假彩色增强就是把一幅由三基色描绘的彩色原图，或者一组描绘同一目标的多光谱图像，逐点映射到由显示三基色所确定的色空间。设这组描述同一目标的多光谱图像为 $f_1(m, n)$, $f_2(m, n)$, …, $f_k(m, n)$，所映射的显示三基色分量的值为 $R(m, n)$, $G(m, n)$, $B(m, n)$，则假彩色增强可定义为：

$$\begin{cases} R(m, n) = T_R[f_1(m, n), \cdots, f_k(m, n)] \\ G(m, n) = T_G[f_1(m, n), \cdots, f_k(m, n)] \\ B(m, n) = T_B[f_1(m, n), \cdots, f_k(m, n)] \end{cases} \tag{4.5-13}$$

式中：k 表示分别在光谱的 k 个不同波段上摄取了 k 幅图像。这时目标上同一像素将对应 k 个原始值，经由式(4.5-13)映射为显示三基色分量的值，故这是"多对三"的映射。T_R, T_G, T_B 是原图像灰度值的线性或非线性函数。假彩色增强不改变原来像素的位置。

运用假彩色增强可使显示图像中的目标具有与原设想不同的颜色。这种增强主要用于：

(1) 把正常的目标置于特定的彩色环境中，使观察者对该目标更加注意，并可防止对新彩色环境的完全适应。

(2) 可使正常目标具有与人眼彩色感觉的灵敏度相匹配的彩色。例如人眼对绿色特别灵敏，因而感兴趣的目标变为绿色就更容易检测出来；人眼对于蓝光变化的对比灵敏度较高，因此可把物体细节映射到蓝色区域，以改善细节的可检测性。

(3) 对于由光谱响应在可见光以外(如红外波段)的传感器所摄取的多光谱图像，则可由假彩色再现这些图像。

当式(4.5-13)表示的是一种线性函数的映射关系时，可用矩阵形式表示这种映射：

$$\begin{bmatrix} R \\ G \\ B \end{bmatrix} = \begin{bmatrix} h_{11} & h_{12}\cdots h_{1k} & C_1 \\ h_{21} & h_{22}\cdots h_{2k} & C_2 \\ h_{31} & h_{32}\cdots h_{3k} & C_3 \end{bmatrix} \begin{bmatrix} f_1 \\ f_2 \\ \vdots \\ f_k \\ 1 \end{bmatrix} \tag{4.5-14}$$

习　题

1. 如原图的直方图为 $p(i)$，灰度 i 的变化范围为 $I_0 \leqslant i \leqslant I_M$，现在把它的直方图

修改为 $p'(j)$，j 的变化范围为 $J_0 \leqslant j \leqslant J_M$。求变换函数 $j = f(i)$。

(1) $p'(j) = \frac{a}{j}$(对数双曲分布)，　$J_0 \leqslant j \leqslant J_M$

(2) $p'(j) = a\exp[-a(j-J_0)]$(指数分布)，　$J_0 \leqslant j$

(3) $p'(j) = a(j-J_0)\exp[-(j-J_0)^2/2a^2]$，　$J_0 \leqslant j$

2. 设一幅数字图像有16级灰度，其直方图分布如下：

$p_0 = 0.03$，$p_1 = 0.00$，$p_2 = 0.06$，$p_3 = 0.10$，$p_4 = 0.20$，$p_5 = 0.11$，

$p_6 = 0.00$，$p_7 = 0.00$，$p_8 = 0.00$，$p_9 = 0.03$，$p_{10} = 0.00$，$p_{11} = 0.06$，

$p_{12} = 0.10$，$p_{13} = 0.20$，$p_{14} = 0.11$，$p_{15} = 0.00$

用 p'_l 去修改直方图，用 $F'_{l-1} < F_k \leqslant F'_l$ 求灰度的对应关系。求处理后的直方图 Q_l，并用草图画出其分布图形。

(1) p'_l 为均匀分布，共10个灰度级。

(2) p'_l 为三角形分布，共11个灰度级。

$p'_0 = 0.00$，$p'_1 = 0.04$，$p'_2 = 0.08$，$p'_3 = 0.12$，$p'_4 = 0.16$，$p'_5 = 0.20$，

$p'_6 = 0.16$，$p'_7 = 0.12$，$p'_8 = 0.08$，$p'_9 = 0.04$，$p'_{10} = 0.00$。

(3) p'_l 分布如下，共16个灰度级。

$p'_0 = 0.00$，$p'_1 = 0.00$，$p'_2 = 0.00$，$p'_3 = 0.03$，$p'_4 = 0.06$，$p'_5 = 0.10$，

$p'_6 = 0.20$，$p'_7 = 0.11$，$p'_8 = 0.03$，$p'_9 = 0.06$，$p'_{10} = 0.10$，$p'_{11} = 0.20$，

$p'_{12} = 0.11$，$p'_{13} = 0.00$，$p'_{14} = 0.00$，$p'_{15} = 0.00$。

3. 用下列模板对图像进行处理，问对应的二维数字滤波器的频率响应是怎样的？它的最大值是多少，在何处？最小值是多少，在何处？是高通滤波器类型还是其他什么类型？

(1) $\begin{bmatrix} 0 & -\beta & 0 \\ -\beta & \alpha+4\beta & -\beta \\ 0 & -\beta & 0 \end{bmatrix}$　(2) $\begin{bmatrix} -\alpha & -\alpha & -\alpha \\ -\alpha & 8\alpha & -\alpha \\ -\alpha & -\alpha & -\alpha \end{bmatrix}$　(3) $\frac{1}{16}\begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix}$

4. 编程实现对一幅灰度图像的非线性对数变换，变换表达式为：

$$g(i, j) = a + \frac{\ln[f(i, j) + 1]}{b\ln c}$$

式中：a，b，c 用于调整曲线的位置和形状。请观察和分析变换前后的图像变化。

5. 设有一幅 32×32 大小的图像，共有8个灰度级，概率分布如下：

$p_0 = 0.10$，$p_1 = 0.20$，$p_2 = 0.16$，$p_3 = 0.05$，$p_4 = 0.17$，

$p_5 = 0.11$，$p_6 = 0.04$，$p_7 = 0.17$。

试对该图像进行直方图均衡化处理，并画出处理前后的直方图。

6. 用中值滤波法对第2章习题3中的 256×256 图像进行内插放大成 512×512 图

像，求在放大后的坐标系(m', n')中m'和n'分别为254、255、256的9个点的灰度值。并与用双线性内插处理后的结果相比较。

7. 用加权中值滤波进行图像内插，如题图4.1所示，问：x位置处的值是多少？

100	·	100	·	0
·	·	·	·	·
100	·	100	x	0
·	·	·	·	·
100	·	0	·	0

题图4.1

参 考 文 献

[1] Gazley C, Reiber J E, Stratton R H. Computer works a new trick in seeing pseudo Color Processing [M]. Aeronautics and Astronautics, 1967.

[2] Castleman K R. 数字图像处理[M]. 朱志刚，等译. 北京：电子工业出版社，1998.

[3] Bovik Ed Al. Handbook of Image and Video Processing[M]. Academic Press, 2000.

[4] Nichols L W, Lamar J. Conversion of infrared images to visible in color[M]. Appl. Opt., 1968.

[5] Kreins E R, Allision L J. Color enhancement of nimbus high resolution infrared radiometer data [M]. Appl. Opt., 1970.

[6] 普拉特 W K. 数字图像处理学[M]. 高荣坤，等译. 北京：科学出版社，1984.

[7] Fink W. Image coloration as an interpretation and. Proc. SPIE/OSA Conference on Image Processing[J]. Pacific Grovo, California, Vlo. 74, 1976, 209～215.

[8] Scher Ann, et al.. Some new image smoothing technique[J]. IEEE Trans. SMC Vol. SMC-10, No. 3, 1980, 153～158.

[9] 冈萨雷斯. 数字图像处理(第二版)[M]. 阮秋琦，等译. 北京：电子工业出版社，2003.

[10] 罗申菲尔特 A，卡克 A C. 数字图像处理[M]. 余英林，等译. 北京：人民邮电出版社，1982.

[11] Lev A, et al.. Interactive enhancement of noisy image[J]. IEEE Trans. SMC. Vol. SMC-7, No. 6, 1977, 435～442.

[12] 周新伦. 局部平均法图像平滑技术综述[J]. 通信学报，1983, 2.

[13] 奥本海姆 A V，谢弗 R W. 数字信号处理[M]. 董士嘉，等译. 北京：科学出版社，1981.

[14] Donoho D L, Adapting to unknown smoothness via wavelet shrinkage[J]. J Amer Statist Assoc, 1995, 90: 1200～1224.

[15] Donoho D L, Johnstone I, Wavelet shrinkage asymptopia[J]. Journal of Royal Statistical Society, 1995, 57(2): 301～369.

[16] Donoho D L, Denoising by soft thresholding[J], IEEE Transaction on Information, 1995, (3): 613～627.

第5章　图像复原与重建

在成像过程中，由于成像系统各种因素的影响，可能使图像质量降低。这种图像质量的降低，在本章中称之为"退化"。与图像增强相似，图像复原的目的也是改善图像的质量。但是图像复原是试图利用退化过程的先验知识使已被退化的图像恢复本来面目，而图像增强是用某种试探的方式改善图像质量，以适应人眼的视觉与心理，这是图像复原和图像增强的本质差别。因此，本章讨论的基础是退化的数学模型，图像的复原可以看成是退化的逆过程。

在成像系统中，引起图像退化的原因很多。例如，成像系统的散焦，成像设备与物体的相对运动，成像器材的固有缺陷以及外部干扰等。由于这些因素造成的图像退化的典型现象是图像模糊，去模糊是一种基本的复原问题。图像复原是基于图像退化的数学模型，复原的方法也建立在比较严格的数学推导上。有较多而复杂的数学运算是本章的特点。

图像重建可被看作一类特殊的图象复原技术。本章将以计算机断层摄像术(Computed Tomography, CT)为例，介绍根据物体横剖面的一组投影数据来重建该横剖面的图像的原理。

5.1　退化的数学模型

设有一成像系统，当输入为 $f(x, y)$ 时，输出为 $g(x, y)$，即

$$g(x, y) = T[f(x, y)] \tag{5.1-1}$$

运算 $T[\cdot]$ 表示这一系统对输入图像的作用，它可能是线性的，也可能是非线性的。现在讨论线性系统的情况。

一幅图像 $f(x, y)$ 可以看做由一系列点源组成的，因此，$f(x, y)$ 可以通过点源函数的积分来表示。

$$f(x, y) = \iint_{-\infty}^{\infty} f(\alpha, \beta)\delta(x-\alpha, y-\beta)\mathrm{d}\alpha\mathrm{d}\beta \tag{5.1-2}$$

式中：δ 函数为点源函数，且有

$$\delta(x-\alpha, y-\beta) = \delta(x-\alpha)\delta(y-\beta) \tag{5.1-3}$$

因此经成像系统处理后，输出图像为

$$\begin{aligned} g(x, y) &= T[f(x, y)] \\ &= \iint_{-\infty}^{\infty} f(\alpha, \beta) T[\delta(x-\alpha, y-\beta)] \mathrm{d}\alpha \mathrm{d}\beta \\ &= \iint_{-\infty}^{\infty} f(\alpha, \beta) h(x, \alpha; y, \beta) \Big] \mathrm{d}\alpha \mathrm{d}\beta \end{aligned} \tag{5.1-4}$$

式中：$h(x, \alpha; y, \beta) = T[\delta(x-\alpha, y-\beta)]$ 称为系统的点扩展函数，或系统的冲击响应。图像退化正是由于系统的冲击响应不是理想的。在通常情况下，系统是移不变的，这时

$$h(x, \alpha; y, \beta) = h(x-\alpha, y-\beta)$$

因此输出图像为

$$g(x, y) = \iint_{-\infty}^{\infty} f(\alpha, \beta) h(x-\alpha, y-\beta) \mathrm{d}\alpha \mathrm{d}\beta = f(x, y) * h(x, y) \tag{5.1-5}$$

如果 $T[\cdot]$是可分离系统，则

$$h(x, \alpha; y, \beta) = h_1(x, \alpha) h_2(y, \beta)$$

一个系统如果满足这个条件，就可以把二维问题化作一维问题，这与正交变换中可分离核是相似的。

对一个线性、移不变、可分离系统，它所成的像可表示为

$$g(x, y) = \iint_{-\infty}^{\infty} f(\alpha, \beta) h_1(x-\alpha) h_2(y-\beta) \mathrm{d}\alpha \mathrm{d}\beta \tag{5.1-6}$$

图像复原就是对某一系统而言，已知 $g(x, y)$，求 $f(x, y)$的问题。

图像退化除了成像系统本身的因素外，有时还受到噪声的污染。一般假定噪声 $n(x, y)$是加性白噪声。这时退化后的图像为

$$g(x, y) = \iint_{-\infty}^{\infty} f(\alpha, \beta) h(x-\alpha, y-\beta) \mathrm{d}\alpha \mathrm{d}\beta + n(x, y) \tag{5.1-7}$$

图 5.1 是图像退化的示意图。

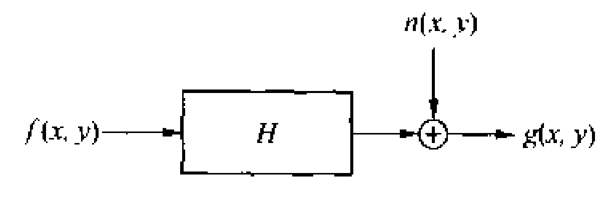


图 5.1　图像的退化模型

5.2 连续系统的图像复原

5.2.1 一般原理

式(5.1-7)所示退化模型可以简写成

$$g = T[f] + n \tag{5.2-1}$$

图像复原就是求上式的解 f。如果有解，则可写成

$$f = T^{-1}[g - n] \tag{5.2-2}$$

对于一个特定的图像来说，其解应该是唯一的。但是可以证明，如果不附加条件，式(5.1-7)的解并不是唯一的。因为我们总可以找到一个函数 $e(x, y)$ 和 $h(x-\alpha, y-\beta)$ 正交，即

$$\iint_{-\infty}^{\infty} h(x-\alpha, y-\beta)e(\alpha, \beta)\mathrm{d}\alpha\mathrm{d}\beta = 0$$

因此，$f(x, y) + ce(x, y)$ 都是这个积分方程的解。所以，解不是唯一的。

另外，在研究傅里叶级数时，有这样一个定理：如果 $\varphi(x)$ 在任意区间内可积，而且

$$\int_{-\infty}^{\infty} |\varphi(x)| \mathrm{d}x < \infty$$

则

$$\lim_{\lambda\to\infty}\int_{-\infty}^{\infty} \varphi(x)\sin(\lambda x)\mathrm{d}x = 0$$

$$\lim_{\lambda\to\infty}\int_{-\infty}^{\infty} \varphi(x)\cos(\lambda x)\mathrm{d}x = 0$$

所以只要 $\iint_{-\infty}^{\infty} |h(x-\alpha, y-\beta)| \mathrm{d}\alpha\mathrm{d}\beta$ 对任意的 x，y 有界，则对任意小的数 ε，总能找到一个数 A，当 λ_1，$\lambda_2 > A$ 时，使积分

$$\iint_{-\infty}^{\infty} h(x-\alpha, y-\beta)\sin(\lambda_1\alpha)\sin(\lambda_2\beta)\mathrm{d}\alpha\mathrm{d}\beta < \varepsilon$$

因此，式(5.1-7)这样的积分方程是没有唯一解的，只能在许多解中找出满足特定条件的合理解[1]。根据不同的特定条件可以形成不同的复原方法，下面将分别进

行讨论。

5.2.2 反滤波

设成像系统是线性移不变的，则

$$g(x, y) = \iint_{-\infty}^{\infty} f(\alpha, \beta) h(x-\alpha, y-\beta) \mathrm{d}\alpha \mathrm{d}\beta + n(x, y) \tag{5.2-3}$$

对式(5.2-3)两边作傅里叶变换，得

$$G(u, v) = F(u, v)H(u, v) + N(u, v) \tag{5.2-4}$$

式中：$G(u, v)$，$F(u, v)$，$H(u, v)$，$N(u, v)$分别是$g(x, y)$，$f(x, y)$，$h(x, y)$，$n(x, y)$的傅里叶变换；$H(u, v)$又称系统的转移函数。

在没有噪声的理想情况下，式(5.2-4)变为

$$G(u, v) = F(u, v)H(u, v) \tag{5.2-5}$$

在上式两边乘以 $\dfrac{1}{H(u, v)}$，得

$$F(u, v) = \frac{G(u, v)}{H(u, v)} \tag{5.2-6}$$

因此如果已知系统的转移函数 $H(u, v)$，用上式即可求得 $F(u, v)$，由此可以通过傅里叶反变换得到原始图像

$$f(x, y) = \mathscr{F}^{-1}\left[\frac{G(u, v)}{H(u, v)}\right] \tag{5.2-7}$$

这种复原方法就称为反滤波。频域的反滤波模型如图5.2所示。

$F(u, v) \longrightarrow$ [$H(u, v)$] $\xrightarrow{G(u, v)}$ [$1/H(u, v)$] $\longrightarrow F(u, v)$

图5.2 频域的反滤波模型(无噪声)

但是，在实际情况下总是有噪声存在的，若仍用反滤波方法，则从式(5.2-4)可得 $F(u, v)$的估值如下式。

$$\hat{F}(u, v) = F(u, v) + \frac{N(u, v)}{H(u, v)} \tag{5.2-8}$$

反滤波复原的原理十分简单，但是在实现中存在较大的困难而使复原效果受到很大影响。首先是 $H(u, v)$零点的影响。在无噪声的理想情况下，如果 $H(u, v)$在

某些对图像信号有较大影响的点或区域上为零，那么 $G(u, v)$ 的值在这些频率处也为零，因此就无法用式(5.2-6)确定这些频率处 $F(u, v)$ 的值。而在存在噪声的一般情况下，从式(5.2-8)可见，估值 $\hat{F}(u, v)$ 和实际值 $F(u, v)$ 之间相差一项 $N(u, v)/H(u, v)$。显然，在 $H(u, v)=0$ 的频率上，由于噪声的存在，使这些频率上的估值 $\hat{F}(u, v)$ 没有意义。而在这些零点附近，由于 $H(u, v)$ 非常小，$N(u, v)/H(u, v)$ 变得很大，从而使误差大大增加，以致无法得到正确的复原结果。

在一般情况下，$H(u, v)$ 具有低通特性，即在原点附近的有限区域内 $H(u, v)\neq 0$，而且迅速下降。而噪声则由于有较宽的带宽，下降的速度要慢得多。因此，反滤波复原常在原点附近的有限区域内进行，以避免 $H(u, v)$ 出现零点或有较小的值。这样可以得到比较合理的结果，但是仍然不可避免地要对复原后的图像质量产生较大的影响。

现在再讨论图像截取所产生的影响。设 $g(x, y)$ 是定义在 $-\infty\leqslant x, y\leqslant\infty$ 上的一张模糊图，$g_1(x, y)$ 是从 $g(x, y)$ 中截取的一部分，即

$$g_1(x, y)=k(x, y)g(x, y) \tag{5.2-9}$$

式中

$$k(x, y)=\begin{cases}1, & x, y\in R\\ 0, & \text{其他}\end{cases}$$

式中：R 为一矩形区域。现在对 $g(x, y)$ 作反滤波，假设没有噪声，则

$$F(u, v)=\frac{G(u, v)}{H(u, v)} \tag{5.2-10}$$

它的反变换为 $f(x, y)$，是定义在 $-\infty\leqslant x, y\leqslant\infty$ 上的正确解，现对 $g_1(x, y)$ 作反滤波，有

$$F_1(u, v)=\frac{G_1(u, v)}{H(u, v)} \tag{5.2-11}$$

由式(5.2-9)可得

$$G_1(u, v)=K(u, v)*G(u, v) \tag{5.2-12}$$

代入式(5.2-11)后得

$$F_1(u, v)=\frac{K(u, v)*G(u, v)}{H(u, v)}\neq K(u, v)*\frac{G(u, v)}{H(u, v)} \tag{5.2-13}$$

从式(5.2-10)和(5.2-13)可见

$$F_1(u, v) \neq K(u, v) * F(u, v)$$

所以
$$f_1(x, y) \neq k(x, y) f(x, y) \tag{5.2-14}$$
即复原以后的图像不是正确解。

5.2.3 维纳滤波

我们仍然考虑线性移不变系统，并重写式(5.2-3)如下

$$g(x, y) = \iint_{-\infty}^{\infty} f(\alpha, \beta) h(x-\alpha, y-\beta) \mathrm{d}\alpha \mathrm{d}\beta + n(x, y) \tag{5.2-15}$$

现在求这一方程的解 $f(x, y)$ 的估值 $\hat{f}(x, y)$，设它有如下形式

$$\hat{f}(x, y) = \iint m(x-\alpha, y-\beta) g(\alpha, \beta) \mathrm{d}\alpha \mathrm{d}\beta \tag{5.2-16}$$

假设 $g(x, y)$（因此也包括 $f(x, y)$ 和 $n(x, y)$）是平稳的随机变量，$h(x, y)$ 是确知的函数。现在求 $m(x, y)$，使误差

$$e^2 = E[\,|f(x, y) - \hat{f}(x, y)|^2] \tag{5.2-17}$$

最小。这就是著名的维纳滤波问题，解法很多。这里直接给出连续的维纳滤波的最后结果。

设 $S_{ff^*}(u, v)$ 为 $f(x, y)$ 的相关函数 $R_{ff^*}(x, y)$ 的傅里叶变换。$S_{gg^*}(u, v)$ 是 $g(x, y)$ 的相关函数 $R_{gg^*}(x, y)$ 的傅里叶变换。$S_{nn^*}(u, v)$ 是噪声 $n(x, y)$ 的相关函数 $R_{nn^*}(u, v)$ 的傅里叶变换。$H(u, v)$ 是点扩展函数 $h(x, y)$ 的傅里叶变换，则有

$$M(u, v) = \frac{H^*(u, v) S_{f^*f}(u, v)}{S_{g^*g}} = \frac{H^*(u, v) S_{f^*f}(u, v)}{H(u, v) H^*(u, v) S_{f^*f}(u, v) + S_{n^*n}(u, v)}$$

$$= \frac{|H(u, v)|^2}{H(u, v)[\,|H(u, v)|^2 + S_{n^*n}(u, v)/S_{f^*f}(u, v)]} \tag{5.2-18}$$

式中：

$$S_{f^*f}(u, v) = S_{ff^*}(-u, -v),\ S_{g^*g}(u, v) = S_{gg^*}(-u, -v),$$

$$S_{n^*n}(u, v) = S_{nn^*}(-u, -v)$$

这就是维纳滤波的解，式中 $S_{n^*n}(u, v)/S_{f^*f}(u, v)$ 是噪声和信号能量之比。若噪声 $n(x, y) = 0$，则上式变为 $M(u, v) = \dfrac{1}{H(u, v)}$，即为反滤波。为了计算 S_{n^*n}，可以令输入 $f(x, y) = 0$，则 $g(x, y) = n(x, y)$，由此计算 $R_{n^*n}(u, v)$，并得

$S_{n^*n}=\mathscr{F}[R_{n^*n}(u,v)]$。

当解 $M(u,v)$满足上式时，均方误差为

$$e^2=\iint\frac{S_{n^*n}(u,v)}{|H(u,v)|^2+S_{n^*n}(u,v)/S_{f^*f}(u,v)}\mathrm{d}u\mathrm{d}v \qquad (5.2-19)$$

5.3　离散情况下的退化模型

前一节讨论了在连续情况下，对一个线性移不变系统而言，退化模型可以用连续函数的卷积来表示。同样，在离散情况下，退化模型可以表示成离散卷积的形式。下面先讨论一维的情况。

5.3.1　一维

为使讨论简化，不考虑噪声的存在，这时退化模型可用下式离散卷积来表示。

设离散序列 $f(m)$定义在 $m=0,1,\cdots,N-1$ 各点上，在其他 m 值上 $f(m)=0$；$h(m)$定义在 $m=0,1,\cdots,M-1$ 各点上，在其他 m 值上 $h(m)=0$；且 $N>M$。则它们的离散卷积为

$$g(m)=\sum_{k=0}^{N-1}f(k)h(m-k)=f(m)*h(m) \qquad (5.3-1)$$

由此得到的结果 $g(m)$在 $m=0,1,\cdots,M+N-2$ 上有确定的值，在其他 m 值上 $g(m)=0$。上式还可以用矩阵的形式表示。

$$\begin{bmatrix} g(0)\\ g(1)\\ \vdots\\ g(M-1)\\ \vdots\\ g(M+N-3)\\ g(M+N-2)\end{bmatrix}=\begin{bmatrix} h(0) & 0 & 0 & 0 & 0\\ h(1) & h(0) & 0 & 0 & 0\\ \vdots & \vdots & \vdots & \vdots & \vdots\\ h(M-1) & h(M-2)\ \cdots & h(0) & 0 & 0\\ 0 & h(M-1)\ \cdots & h(1) & \vdots & \vdots\\ \vdots & 0 & \vdots & \vdots & \vdots\\ \vdots & \vdots & \vdots & \vdots & \vdots\\ 0 & 0 & 0 & h(M-1) & h(M-2)\\ 0 & 0 & 0 & 0 & h(M-1)\end{bmatrix}\begin{bmatrix} f(0)\\ f(1)\\ \vdots\\ \vdots\\ \\ \\ f(N-1)\end{bmatrix}$$

(5.3-2)

式中：$[\boldsymbol{h}]$是 $(M-N-1)\times N$ 阶矩阵，它的每一行都是 $h(M-1),h(M-2),\cdots,h(1),h(0)$ 逐步向右移动而成。

离散卷积还可用周期卷积来表示。首先将 $f(m)$ 和 $h(m)$ 分别作周期延拓，周期都为 $P=M+N-1$，即在一个周期内

$$f_e(m)=\begin{cases}f(m), & 0\leqslant m\leqslant N-1\\ 0, & N\leqslant m\leqslant P-1\end{cases}$$

$$h_e(m)=\begin{cases}h(m), & 0\leqslant m\leqslant M-1\\ 0, & M\leqslant m\leqslant P-1\end{cases}$$

周期延拓是为了避免卷积的各个周期产生交迭。则周期卷积可定义为

$$f_e(m) \circledast h_e(m)=\sum_{k=0}^{p-1} f_e(k)h_e((m-k))_p \tag{5.3-3}$$

式中：m 为任意值，$((k))_p = k \bmod p$，在一个周期内

$$f_e(m) \circledast h_e(m)=f(m)*h(m)=g(m)$$

即

$$g(m)=\sum_{k=0}^{p-1} f_e(k)h_e(m-k),\ m=0,1,\cdots,P-1 \tag{5.3-4}$$

用矩阵表示，则为

$$\begin{bmatrix} g(0)\\ g(1)\\ \vdots\\ g(P-1)\end{bmatrix}=\begin{bmatrix} h_e(0) & h_e(-1) & h_e(-2) & \cdots h_e(-P+1)\\ h_e(1) & h_e(0) & h_e(-1) & \cdots h_e(-P+2)\\ \vdots & \vdots & \vdots & \vdots\\ h_e(P-1) & h_e(P-2) & h_e(P-3) & \cdots h_e(0)\end{bmatrix}\begin{bmatrix} f_e(0)\\ f_e(1)\\ \vdots\\ f_e(P-1)\end{bmatrix} \tag{5.3-5}$$

因为 $h_e(m)$ 的周期为 P，所以 $h_e(m+P)=h_e(m)$，因此

$$\begin{bmatrix} g(0)\\ g(1)\\ \vdots\\ g(P-1)\end{bmatrix}=\begin{bmatrix} h_e(0) & h_e(P-1) & h_e(P-2) & \cdots h_e(1)\\ h_e(1) & h_e(0) & h_e(P-1) & \cdots h_e(2)\\ \vdots & \vdots & \vdots & \vdots\\ h_e(P-1) & h_e(P-2) & h_e(P-3) & \cdots h_e(0)\end{bmatrix}\begin{bmatrix} f_e(0)\\ f_e(1)\\ \vdots\\ f_e(P-1)\end{bmatrix} \tag{5.3-6}$$

还可写成更简洁的形式

$$\boldsymbol{g}=[\boldsymbol{h}]\boldsymbol{f} \tag{5.3-7}$$

式中：$\boldsymbol{g}$ 和 $\boldsymbol{f}$ 都是 p 维列向量；$[\boldsymbol{h}]$ 则是 $p\times p$ 阶矩阵。

从式(5.3-6)可见，矩阵 $[\boldsymbol{h}]$ 中每一行元素均相同，只是每行以循环方式右移

一位，因此矩阵[$\boldsymbol{h}$]称为循环矩阵。可以证明，循环矩阵相加还是循环矩阵；循环矩阵相乘也还是循环矩阵。

现在研究矩阵方程(5.3-7)在傅里叶变换下的情况。设向量$\boldsymbol{G}$和向量$\boldsymbol{F}$是向量$\boldsymbol{g}$和向量$\boldsymbol{f}$的傅里叶变换，则有

$$\boldsymbol{F} = [\boldsymbol{U}]\boldsymbol{f},\ \boldsymbol{G} = [\boldsymbol{U}]\boldsymbol{g} \tag{5.3-8}$$

式中[$\boldsymbol{U}$]为变换矩阵

$$[\boldsymbol{U}] = \frac{1}{\sqrt{P}}\begin{bmatrix} 1 & 1 & 1 & \cdots & 1 \\ 1 & W_P^1 & W_P^2 & \cdots & W_P^{P-1} \\ 1 & W_P^2 & W_P^4 & \cdots & W_P^{2(P-1)} \\ \vdots & \vdots & \vdots & & \vdots \\ 1 & W_P^{P-1} & W_P^{2(P-1)} & \cdots & W_P^{(P-1)^2} \end{bmatrix} = [\boldsymbol{u}_1 \quad \boldsymbol{u}_2 \quad \cdots \quad \boldsymbol{u}_{P-1}] \tag{5.3-9}$$

这里

$$W_P = \exp(-\mathrm{j}2\pi/P)$$

对式(5.3-7)两边作傅里叶变换，并利用[$\boldsymbol{U}$]的酉矩阵性质$[\boldsymbol{U}]*[\boldsymbol{U}] = \boldsymbol{I}$，有

$$\boldsymbol{G} = [\boldsymbol{U}]\boldsymbol{g} = [\boldsymbol{U}][\boldsymbol{h}]\boldsymbol{f} = [\boldsymbol{U}][\boldsymbol{h}][\boldsymbol{U}]^*[\boldsymbol{U}]\boldsymbol{f} = [\boldsymbol{U}][\boldsymbol{h}][\boldsymbol{U}]^*\boldsymbol{F} \tag{5.3-10}$$

先看$[\boldsymbol{h}][\boldsymbol{U}]^*$中的某一列

$$[\boldsymbol{h}]\boldsymbol{u}_j^* = \frac{1}{\sqrt{P}}\begin{bmatrix} h_e((P))_p & h_e((P-1))_p & \cdots & h_e((1))_p \\ h_e((P+1))_p & h_e((P))_p & \cdots & h_e((2))_p \\ \vdots & \vdots & & \vdots \\ h_e((2P-1))_p & h_e((2P-2))_p & \cdots & h_e((P))_p \end{bmatrix}\begin{bmatrix} W_P^0 \\ W_P^{-j} \\ W_P^{-2j} \\ \vdots \\ W_P^{-(P-1)j} \end{bmatrix}$$

$$= \frac{1}{\sqrt{P}}\begin{bmatrix} \sum_{i=1}^{P} h_e((i))_p W_P^{-j(P-i)} \\ \sum_{i=2}^{P+1} h_e((i))_p W_P^{-j(P+1-i)} \\ \vdots \\ \sum_{i=P}^{2P-1} h_e((i))_p W_P^{-j(2P-1-i)} \end{bmatrix} = \frac{1}{\sqrt{P}}\begin{bmatrix} \sum_{i=0}^{P-1} h_e(i) W_P^{ji} \\ \sum_{i=0}^{P-1} h_e(i) W_P^{ji-j} \\ \vdots \\ \sum_{i=0}^{P-1} h_e(i) W_P^{ji-(P-1)j} \end{bmatrix}$$

$$= \frac{1}{\sqrt{P}} \sum_{i=0}^{P-1} h_e(i) W_P^{ij} \begin{bmatrix} W_P^0 \\ W_P^{-j} \\ \vdots \\ W_P^{-(P-1)j} \end{bmatrix} = \lambda_h(j) \boldsymbol{u}_j^* \tag{5.3-11}$$

从上式可以看出，因为

$$[\boldsymbol{h}]\boldsymbol{u}_j^* = \lambda_h(j)\boldsymbol{u}_j^*$$

所以 $\lambda_h(j)$ 是 $[\boldsymbol{h}]$ 的第 j 个特征值，而且

$$\lambda_h(j) = \sum_{i=0}^{P-1} h_e(i) W_P^{ij} = H(j) \tag{5.3-12}$$

即特征值 $\lambda_h(j)$ 是 $h_e(i)$ 的离散傅里叶变换。

由式(5.3-11)和式(5.3-12)可以得到

$$[\boldsymbol{h}][\boldsymbol{U}]^* = [\boldsymbol{U}]^* \begin{bmatrix} H(0) & 0 & \cdots & 0 \\ 0 & H(1) & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & H(P-1) \end{bmatrix} = [\boldsymbol{U}]^* [\boldsymbol{\Lambda}_h]$$

式中：$[\boldsymbol{\Lambda}_h]$ 为对角阵

$$[\boldsymbol{\Lambda}_h] = \begin{bmatrix} H(0) & 0 & \cdots & 0 \\ 0 & H(1) & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & H(P-1) \end{bmatrix}$$

由式(5.3-10)得

$$\boldsymbol{G} = [\boldsymbol{U}][\boldsymbol{h}][\boldsymbol{U}]^* \boldsymbol{F} = [\boldsymbol{U}][\boldsymbol{U}]^* [\boldsymbol{\Lambda}_h]\boldsymbol{F} = [\boldsymbol{\Lambda}_h]\boldsymbol{F} \tag{5.3-13}$$

即有

$$\begin{bmatrix} G(0) \\ G(1) \\ \vdots \\ G(P-1) \end{bmatrix} = \begin{bmatrix} H(0) & 0 & \cdots & 0 \\ 0 & H(1) & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & H(P-1) \end{bmatrix} \begin{bmatrix} F(0) \\ F(1) \\ \vdots \\ F(P-1) \end{bmatrix}$$

所以对每一个分量都有

$$G(k) = H(k)F(k) \tag{5.3-14}$$

这一结果说明，循环卷积的傅里叶变换就是卷积函数的傅里叶变换的积。

5.3.2 二维

设一离散二维信号 $f(m, n)$，它在 $M\times N$ 大小的范围内有值，其他位置上为零，即

$$f(m, n)=\begin{cases}f(m, n), & 0\leqslant m\leqslant M-1 \text{ 且 } 0\leqslant n\leqslant N-1\\ 0, & \text{其他}\end{cases}$$

且设

$$h(m, n)=\begin{cases}h(m, n), & 0\leqslant m\leqslant J-1 \text{ 且 } 0\leqslant n\leqslant K-1\\ 0, & \text{其他}\end{cases}$$

这时二维卷积可以写成

$$g(m, n)=\sum_{k=0}^{M-1}\sum_{l=0}^{N-1}h(m-k, n-l)f(k, l)=h(m, n)*f(m, n) \tag{5.3-15}$$

结果 $g(m, n)$ 在 $P\times Q$ 范围内有值，$P=M+J-1$，$Q=N+K-1$，即有

$$g(m, n)=\begin{cases}g(m, n), & 0\leqslant m\leqslant P-1 \text{ 且 } 0\leqslant n\leqslant Q-1\\ 0, & \text{其他}\end{cases}$$

这说明卷积结果不为零的区域相当于在 f 的两个边上加上宽度为 $J-1$ 和 $K-1$ 的区域，如图 5.3 所示。

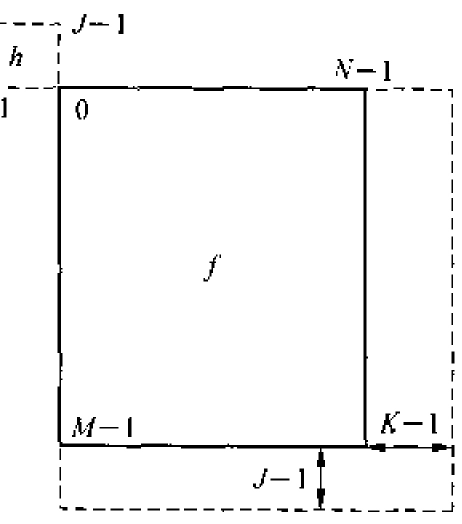


图 5.3 卷积区域示意图

和一维情况一样，也可以用二维周期卷积来代表二维的线性卷积。这时同样要对 $f(m, n)$ 和 $g(m, n)$ 作周期延拓，延拓后的大小为 $P\times Q$，同样有

$$P=M+J-1, \quad Q=N+K-1 \tag{5.3-16}$$

即

$$f_e(m, n)=\begin{cases}f(m, n), & 0\leqslant m\leqslant M-1 \text{ 且 } 0\leqslant n\leqslant N-1\\ 0, & M\leqslant m\leqslant P-1 \text{ 或 } N\leqslant n\leqslant Q-1\end{cases}$$

$$h_e(m, n) = \begin{cases} h(m, n), & 0 \leqslant m \leqslant J-1 \text{ 且 } 0 \leqslant n \leqslant K-1 \\ 0, & J \leqslant m \leqslant P-1 \text{ 或 } K \leqslant n \leqslant Q-1 \end{cases}$$

则 $f_e(m, n)$和 $h_e(m, n)$的周期卷积为

$$\begin{aligned} g_e(m, n) &= \sum_{k=0}^{P-1} \sum_{l=0}^{Q-1} h_e[((m-k))_P, ((n-l))_Q] f_e(k, l) \\ &= h_e(m, n) \circledast f_e(m, n) \end{aligned} \tag{5.3-17}$$

式中:m、n 为任意值。在一个周期内,即当 $0 \leqslant m \leqslant P-1$ 和 $0 \leqslant n \leqslant Q-1$,周期卷积的结果与线性卷积相同,即

$$g_e(m, n) = h(m, n) * f(m, n) = g(m, n) \tag{5.3-18}$$

现在用向量 $\boldsymbol{g}$ 和 $\boldsymbol{f}$ 来表示二维离散图像 $g(m, n)$和 $f(m, n)$。方法是将 $g(m, n)$和 $f(m, n)$中元素按行堆砌成列向量,即

$$\boldsymbol{g} = [g(0, 0) g(0, 1) \cdots g(0, Q-1) g(1, 0) g(1, 1) \cdots g(1, Q-1) \cdots g(P-1, Q-1)]^T$$

若它的第 r 个分量对应于 $g(m, n)$,则有

$$r = Qm + n, \quad 0 \leqslant r \leqslant PQ-1 \tag{5.3-19}$$

用同样的方法堆砌成 f,这样,就可以把式(5.3-17)写成矩阵形式

$$\boldsymbol{g} = [\boldsymbol{h}] \boldsymbol{f} \tag{5.3-20}$$

式中:$[\boldsymbol{h}]$是 $P \times Q$ 矩阵,可以写成

$$[\boldsymbol{h}] = \begin{bmatrix} [\boldsymbol{h}_0] & [\boldsymbol{h}_{P-1}] & [\boldsymbol{h}_{P-2}] & \cdots & [\boldsymbol{h}_1] \\ [\boldsymbol{h}_1] & [\boldsymbol{h}_0] & [\boldsymbol{h}_{P-1}] & \cdots & [\boldsymbol{h}_2] \\ [\boldsymbol{h}_2] & [\boldsymbol{h}_1] & [\boldsymbol{h}_0] & \cdots & [\boldsymbol{h}_3] \\ \vdots & \vdots & \vdots & & \vdots \\ [\boldsymbol{h}_{P-1}] & [\boldsymbol{h}_{P-2}] & [\boldsymbol{h}_{P-3}] & \cdots & [\boldsymbol{h}_0] \end{bmatrix} \tag{5.3-21}$$

式中:$[\boldsymbol{h}_j]$为子矩阵,大小为 $Q \times Q$。即$[\boldsymbol{h}]$矩阵由 $P \times P$ 个大小为$Q \times Q$ 的子矩阵组成。子矩阵的下标是以循环方式排列的,其中

$$[\boldsymbol{h}_j] = \begin{bmatrix} h_e(j, 0) & h_e(j, Q-1) & h_e(j, Q-2) & \cdots & h_e(j, 1) \\ h_e(j, 1) & h_e(j, 0) & h_e(j, Q-1) & \cdots & h_e(j, 2) \\ h_e(j, 2) & h_e(j, 1) & h_e(j, 0) & \cdots & h_e(j, 3) \\ \vdots & \vdots & \vdots & & \vdots \\ h_e(j, Q-1) & h_e(j, Q-2) & h_e(j, Q-3) & \cdots & h_e(j, 0) \end{bmatrix} \tag{5.3-22}$$

显然,矩阵$[\boldsymbol{h}_j]$中元素的第二个下标也是按循环方式变化的,所以$[\boldsymbol{h}_j]$也是循环矩阵。因此,矩阵$[\boldsymbol{h}]$被称为分块循环矩阵。

现在用二维离散傅里叶变换来讨论二维卷积。先讨论向量 $\boldsymbol{g}$ 的傅里叶变换。设 $g(m, n)$的傅里叶变换为 $G(u, v)$,则

$$G(u, v) = \frac{1}{\sqrt{PQ}} \sum_{m=0}^{P-1} \sum_{n=0}^{Q-1} g(m, n) \exp\left[-j2\pi\left(\frac{um}{P} + \frac{vn}{Q}\right)\right] \quad (5.3-23)$$

也可写成矩阵形式

$$[\boldsymbol{G}] = [\boldsymbol{U}][\boldsymbol{g}][\boldsymbol{V}] \quad (5.3-24)$$

$[\boldsymbol{g}]$是图像 $g(m, n)$的矩阵表示,$[\boldsymbol{U}]$、$[\boldsymbol{V}]$为变换矩阵,

$$[\boldsymbol{U}] = \frac{1}{\sqrt{P}} \begin{bmatrix} 1 & 1 & 1 & \cdots & 1 \\ 1 & W_P & W_P^2 & \cdots & W_P^{P-1} \\ 1 & W_P^2 & W_P^4 & \cdots & W_P^{2(P-1)} \\ \vdots & \vdots & \vdots & & \vdots \\ 1 & W_P^{P-1} & W_P^{2(P-1)} & \cdots & W_P^{(P-1)^2} \end{bmatrix}$$

$$[\boldsymbol{V}] = \frac{1}{\sqrt{Q}} \begin{bmatrix} 1 & 1 & 1 & \cdots & 1 \\ 1 & W_Q & W_Q^2 & \cdots & W_Q^{Q-1} \\ 1 & W_Q^2 & W_Q^4 & \cdots & W_Q^{2(Q-1)} \\ \vdots & \vdots & \vdots & & \vdots \\ 1 & W_Q^{Q-1} & W_Q^{2(Q-1)} & \cdots & W_Q^{(Q-1)^2} \end{bmatrix}$$

其中

$$W_P = \exp[-j2\pi/P], \ W_Q = \exp[-j2\pi/Q]$$

同样可以定义$[\boldsymbol{f}]$的傅里叶变换为

$$[\boldsymbol{F}] = [\boldsymbol{U}][\boldsymbol{f}][\boldsymbol{V}] \quad (5.3-25)$$

为了用图像的向量表示进行运算,要用到矩阵的直积(Kronecker 积)的概念(详见第三章 3.1.5 节)。引入了矩阵直积概念后,由(3.1-35)式可以把式(5.3-24)写成

$$\boldsymbol{G} = \{[\boldsymbol{U}] \otimes [\boldsymbol{V}]\}\boldsymbol{g} \quad (5.3-26)$$

式中:$\boldsymbol{G}$, $\boldsymbol{g}$ 都是按行堆砌后的 PQ 维列向量,$[\boldsymbol{U}] \otimes [\boldsymbol{V}]$ 是 $PQ \times PQ$ 阶矩阵。同样 $\boldsymbol{f}$ 的傅里叶变换也可表示成

$$\boldsymbol{F} = \{[\boldsymbol{U}] \otimes [\boldsymbol{V}]\}\boldsymbol{f} \tag{5.3-27}$$

由变换矩阵$[\boldsymbol{U}]$和$[\boldsymbol{V}]$以及它们的直积的性质，有

$$[\boldsymbol{U}]^*[\boldsymbol{U}] = [\boldsymbol{U}][\boldsymbol{U}]^* = \boldsymbol{I}_P \tag{5.3-28a}$$

$$[\boldsymbol{V}]^*[\boldsymbol{V}] = [\boldsymbol{V}][\boldsymbol{V}]^* = \boldsymbol{I}_Q \tag{5.3-28b}$$

$$\{[\boldsymbol{U}]^* \otimes [\boldsymbol{V}]^*\}\{[\boldsymbol{U}] \otimes [\boldsymbol{V}]\} = \boldsymbol{I}_P \otimes \boldsymbol{I}_Q = \boldsymbol{I}_{PQ} \tag{5.3-28c}$$

$$\{[\boldsymbol{U}] \otimes [\boldsymbol{V}]\}\{[\boldsymbol{U}]^* \otimes [\boldsymbol{V}]^*\} = \boldsymbol{I}_P \otimes \boldsymbol{I}_Q = \boldsymbol{I}_{PQ} \tag{5.3-28d}$$

$$[\boldsymbol{U}]^T = [\boldsymbol{U}],\ [\boldsymbol{V}]^T = [\boldsymbol{V}] \tag{5.3-28e}$$

利用式(5.3-26)和式(5.3-28c)得

$$\begin{aligned}\boldsymbol{G} &= \{[\boldsymbol{U}] \otimes [\boldsymbol{V}]\}\boldsymbol{g} = \{[\boldsymbol{U}] \otimes [\boldsymbol{V}]\}[\boldsymbol{h}]\boldsymbol{f} \\ &= \{[\boldsymbol{U}] \otimes [\boldsymbol{V}]\}[\boldsymbol{h}]\{[\boldsymbol{U}]^* \otimes [\boldsymbol{V}]^*\}\{[\boldsymbol{U}] \otimes [\boldsymbol{V}]\}\boldsymbol{f} \\ &= \{[\boldsymbol{U}] \otimes [\boldsymbol{V}]\}[\boldsymbol{h}]\{[\boldsymbol{U}]^* \otimes [\boldsymbol{V}]^*\}\boldsymbol{F}\end{aligned} \tag{5.3-29}$$

和一维情况一样，研究$[\boldsymbol{h}]\{[\boldsymbol{U}]^* \otimes [\boldsymbol{V}]^*\}$的元素

$$\begin{aligned}&([\boldsymbol{h}]\{[\boldsymbol{U}]^* \otimes [\boldsymbol{V}]^*\})_{mn,ij} \\ &= \sum_{k=0}^{P-1}\sum_{l=0}^{Q-1} h_{mn,kl}\{[\boldsymbol{U}]^* \otimes [\boldsymbol{V}]^*\}_{kl,ij} \\ &= \sum_{k=0}^{P-1}\sum_{l=0}^{Q-1} h_e(((m-k))_P,\ ((n-l))_Q)u_{ki}^* u_{lj}^* \\ &= \frac{1}{\sqrt{PQ}}\sum_{k=0}^{P-1}\sum_{l=0}^{Q-1} h_e(((P+m-k))_P,\ ((Q+n-l))_Q)W_P^{-ki} W_Q^{lj} \\ &= \frac{1}{\sqrt{PQ}}\sum_{k'=m+1}^{P-m}\ \sum_{l'=n+1}^{Q+n} h_e(((k'))_P((l'))_Q)W^{-(P+m-k')i}W^{-(Q+n-l')j}\end{aligned}$$

由于k'和l'分别以P和Q为周期，且$W_P^{-Pi} = W_Q^{Qj} = 1$，所以

$$\begin{aligned}&([\boldsymbol{h}]\{[\boldsymbol{U}]^* \otimes [\boldsymbol{V}]^*\})_{mn,ij} \\ &= \frac{1}{\sqrt{PQ}}\sum_{k'=0}^{P-1}\sum_{l'=0}^{Q-1} h_e(k',\ l')W_P^{k'i}W_Q^{l'j}W_P^{-mi}W_Q^{nj} \\ &= H(i,\ j)[\boldsymbol{U}]_{mi}^*[\boldsymbol{V}]_{nj}^* = H(i,\ j)\{[\boldsymbol{U}]^* \otimes [\boldsymbol{V}]^*\}_{mn,ij}\end{aligned} \tag{5.3-30}$$

式中$H(i,\ j)$是$h_e(k,\ l)$的傅里叶变换，即

$$H(i,\ j) = \frac{1}{\sqrt{PQ}}\sum_{k=0}^{P-1}\sum_{n=0}^{Q-1} h_e(m,\ n)W_P^{mi}W_Q^{nj}$$

由式(5.3-30)可得

$$[\boldsymbol{h}]\{[\boldsymbol{U}]^{*}\otimes[\boldsymbol{V}]^{*}\}=\{[\boldsymbol{U}]^{*}\otimes[\boldsymbol{V}]^{*}\}\begin{bmatrix}H(0,0) & & & & & & \\ & H(0,1) & & & & & \\ & & \ddots & & & & \\ & & & H(0,Q-1) & & & \\ & & & & H(1,0) & & \\ & & & & & \ddots & \\ & & & & & & H(P-1,Q-1)\end{bmatrix} \tag{5.3-31}$$

由式(5.3-29)和式(5.3-28d),得到

$$[\boldsymbol{G}]=\begin{bmatrix}H(0,0) & & & & & & \\ & H(0,1) & & & & & \\ & & \ddots & & & & \\ & & & H(0,Q-1) & & & \\ & & & & H(1,0) & & \\ & & & & & \ddots & \\ & & & & & & H(P-1,Q-1)\end{bmatrix}[\boldsymbol{F}] \tag{5.3-32}$$

所以对每一个分量有

$$G(u,v)=H(u,v)F(u,v),\ u=0,1,\cdots,P-1;\ v=0,1,\cdots,Q-1 \tag{5.3-33}$$

就是说,在二维情况下循环卷积的傅里叶变换还是卷积函数的傅里叶变换的乘积。

5.4 离散情况下的复原

这一节讨论离散情况下的复原方法。这些方法都建立在前面讨论的退化模型的基础上,它们的基本思想都是设法找出原始图像的估计$\hat{f}$,使预先确定的某个优化准则最小。这些方法可分为有约束条件的复原和无约束条件的复原。

5.4.1 无约束条件复原

1. 反滤波

前面已经证明，成像系统的输出图像和输入图像的关系可以用堆砌以后的向量方程来表示

$$\boldsymbol{g}=[\boldsymbol{h}]\boldsymbol{f}+\boldsymbol{n} \tag{5.4-1}$$

此式比式(5.3-20)多了一个噪声项 $\boldsymbol{n}$，这是考虑到实际情况而加上的。式中$[\boldsymbol{h}]$为一分块循环矩阵。现在求一个估值向量$\hat{\boldsymbol{f}}$，使 $\boldsymbol{g}-[\boldsymbol{h}]\hat{\boldsymbol{f}}$的幅值最小，或看做使噪声最小。为此定准则函数为

$$\begin{aligned}J[\hat{\boldsymbol{f}}]&=[\boldsymbol{g}-[\boldsymbol{h}]\hat{\boldsymbol{f}}]^{\mathrm{T}}[\boldsymbol{g}-[\boldsymbol{h}]\hat{\boldsymbol{f}}]\\&=\boldsymbol{g}^{\mathrm{T}}\boldsymbol{g}-\hat{\boldsymbol{f}}^{\mathrm{T}}[\boldsymbol{h}]^{\mathrm{T}}\boldsymbol{g}-\boldsymbol{g}^{\mathrm{T}}[\boldsymbol{h}]\hat{\boldsymbol{f}}+\hat{\boldsymbol{f}}^{\mathrm{T}}[\boldsymbol{h}]^{\mathrm{T}}[\boldsymbol{h}]\hat{\boldsymbol{f}}\end{aligned} \tag{5.4-2}$$

为了求得$\hat{\boldsymbol{f}}$，使准则函数最小，可将上式对向量$\hat{\boldsymbol{f}}$求导，并令其等于0。下面用到对向量求导的两个性质，设 $\boldsymbol{a}$ 和 $\boldsymbol{b}$ 为两个向量，$[\boldsymbol{A}]$为对称矩阵，J 为一标量，则

(1) 如果 $J=\boldsymbol{a}^{\mathrm{T}}\boldsymbol{b}=\boldsymbol{b}^{\mathrm{T}}\boldsymbol{a}$，则

$$\frac{\partial J}{\partial \boldsymbol{a}}=\boldsymbol{b} \tag{5.4-3}$$

(2) 如果 $J=\boldsymbol{a}^{\mathrm{T}}[\boldsymbol{A}]\boldsymbol{a}$，则

$$\frac{\partial J}{\partial \boldsymbol{a}}=2[\boldsymbol{A}]\boldsymbol{a} \tag{5.4-4}$$

现对式(5.4-2)求导，并令其等于0，有

$$\frac{\partial J}{\partial \hat{\boldsymbol{f}}}=-[\boldsymbol{h}]^{\mathrm{T}}\boldsymbol{g}-[\boldsymbol{g}^{\mathrm{T}}[\boldsymbol{h}]]^{\mathrm{T}}+2[\boldsymbol{h}]^{\mathrm{T}}[\boldsymbol{h}]\hat{\boldsymbol{f}}=0$$

$$-[\boldsymbol{h}]^{\mathrm{T}}\boldsymbol{g}+[\boldsymbol{h}]^{\mathrm{T}}[\boldsymbol{h}]\hat{\boldsymbol{f}}=0 \tag{5.4-5}$$

若$[\boldsymbol{h}]$的逆$[\boldsymbol{h}]^{-1}$存在，则

$$\{[\boldsymbol{h}]^{\mathrm{T}}[\boldsymbol{h}]\}^{-1}=[\boldsymbol{h}]^{-1}\{[\boldsymbol{h}]^{\mathrm{T}}\}^{-1}$$

代入式(5.4-5)得

$$\hat{\boldsymbol{f}}=[\boldsymbol{h}]^{-1}\{[\boldsymbol{h}]^{\mathrm{T}}\}^{-1}[\boldsymbol{h}]^{\mathrm{T}}\boldsymbol{g}=[\boldsymbol{h}]^{-1}\boldsymbol{g}$$

或

$$\boldsymbol{g}=[\boldsymbol{h}]\hat{\boldsymbol{f}}$$

对上式作傅里叶变换，由5.3.2小节中讨论可知，变换结果为

$$G(u,\ v) = H(u,\ v)\hat{F}(u,\ v)$$

由此得离散情况下的反滤波公式

$$\hat{F}(u,\ v) = G(u,\ v)/H(u,\ v) \tag{5.4-6}$$

这是不考虑噪声情况下的反滤波公式，它与连续情况下的结果有相同的形式。

2. 维纳滤波

已知退化模型的向量表示为

$$\boldsymbol{g} = [\boldsymbol{h}]\boldsymbol{f} + \boldsymbol{n}$$

设解为估值$\hat{\boldsymbol{f}}$，它可表示为

$$\hat{\boldsymbol{f}} = [\boldsymbol{L}]\boldsymbol{g} \tag{5.4-7}$$

现在要求 $PQ \times PQ$ 矩阵$[\boldsymbol{L}]$误差 e^2 最小

$$e^2 = E[\|\boldsymbol{f} - \hat{\boldsymbol{f}}\|^2] \tag{5.4-8}$$

因为

$$\|\boldsymbol{f} - \hat{\boldsymbol{f}}\|^2 = T_r[(\boldsymbol{f} - \hat{\boldsymbol{f}})(\boldsymbol{f} - \hat{\boldsymbol{f}})^{\mathrm{T}}] = T_r[\boldsymbol{f}\boldsymbol{f}^{\mathrm{T}} - \boldsymbol{f}\hat{\boldsymbol{f}}^{\mathrm{T}} - \hat{\boldsymbol{f}}\boldsymbol{f}^{\mathrm{T}} + \hat{\boldsymbol{f}}\hat{\boldsymbol{f}}^{\mathrm{T}}]$$

所以误差 e^2 可表达为

$$e^2 = T_r[E(\boldsymbol{f}\boldsymbol{f}^{\mathrm{T}}) - E(\boldsymbol{f}\hat{\boldsymbol{f}}^{\mathrm{T}}) - E(\hat{\boldsymbol{f}}\boldsymbol{f}^{\mathrm{T}}) + E(\hat{\boldsymbol{f}}\hat{\boldsymbol{f}}^{\mathrm{T}})] \tag{5.4-9}$$

由式(5.4-7)有

$$\hat{\boldsymbol{f}}^{\mathrm{T}} = \boldsymbol{g}^{\mathrm{T}}[\boldsymbol{L}]^{\mathrm{T}} = (\boldsymbol{f}^{\mathrm{T}}[\boldsymbol{h}]^{\mathrm{T}} + \boldsymbol{n}^{\mathrm{T}})[\boldsymbol{L}]^{\mathrm{T}}$$

所以有

$$\boldsymbol{f}\hat{\boldsymbol{f}}^{\mathrm{T}} = \boldsymbol{f}\boldsymbol{f}^{\mathrm{T}}[\boldsymbol{h}]^{\mathrm{T}}[\boldsymbol{L}]^{\mathrm{T}} + \boldsymbol{f}\boldsymbol{n}^{\mathrm{T}}[\boldsymbol{L}]^{\mathrm{T}}$$

$$\hat{\boldsymbol{f}}\boldsymbol{f}^{\mathrm{T}} = [\boldsymbol{L}]\boldsymbol{g}\boldsymbol{f}^{\mathrm{T}} = [\boldsymbol{L}]\{[\boldsymbol{h}]\boldsymbol{f} + \boldsymbol{n}\}\boldsymbol{f}^{\mathrm{T}} = [\boldsymbol{L}][\boldsymbol{h}]\boldsymbol{f}\boldsymbol{f}^{\mathrm{T}} + [\boldsymbol{L}]\boldsymbol{n}\boldsymbol{f}^{\mathrm{T}}$$

$$\begin{aligned}\hat{\boldsymbol{f}}\hat{\boldsymbol{f}}^{\mathrm{T}} &= [\boldsymbol{L}]\boldsymbol{g}\,\hat{\boldsymbol{f}}^{\mathrm{T}} = [\boldsymbol{L}]\{[\boldsymbol{h}]\boldsymbol{f} + \boldsymbol{n}\}\{(\boldsymbol{f}^{\mathrm{T}}[\boldsymbol{h}]^{\mathrm{T}} + \boldsymbol{n}^{\mathrm{T}})[\boldsymbol{L}]^{\mathrm{T}}\}\\ &= [\boldsymbol{L}]\{[\boldsymbol{h}]\boldsymbol{f}\boldsymbol{f}^{\mathrm{T}}[\boldsymbol{h}]^{\mathrm{T}} + [\boldsymbol{h}]\boldsymbol{f}\boldsymbol{n}^{\mathrm{T}} + \boldsymbol{n}\boldsymbol{f}^{\mathrm{T}}[\boldsymbol{h}]^{\mathrm{T}} + \boldsymbol{n}\boldsymbol{n}^{\mathrm{T}}\}[\boldsymbol{L}]^{\mathrm{T}}\end{aligned}$$

设 $\boldsymbol{f}$(实数向量)的自相关矩阵 $\boldsymbol{R}_f$ 和噪声 $\boldsymbol{n}$(实向量)的自相关矩阵 $\boldsymbol{R}_n$ 分别如下：

$$[\boldsymbol{R}_f] = E[\boldsymbol{f}\boldsymbol{f}^{\mathrm{T}}] \tag{5.4-10}$$

$$[\boldsymbol{R}_n] = E[\boldsymbol{n}\boldsymbol{n}^{\mathrm{T}}] \tag{5.4-11}$$

设噪声 $\boldsymbol{n}$ 与图像 $\boldsymbol{f}$ 相互独立，且 $E(\boldsymbol{n}) = 0$，则有

$$E[\boldsymbol{f}\boldsymbol{n}^{\mathrm{T}}] = E[\boldsymbol{f}]E[\boldsymbol{n}^{\mathrm{T}}] = 0$$

由此可得

$$E[\boldsymbol{f}\hat{\boldsymbol{f}}^{\mathrm{T}}] = [\boldsymbol{R}_f][\boldsymbol{h}]^{\mathrm{T}}[\boldsymbol{L}]^{\mathrm{T}}$$

$$E[\hat{\boldsymbol{f}}\boldsymbol{f}^{\mathrm{T}}] = [\boldsymbol{L}][\boldsymbol{h}][\boldsymbol{R}_f]$$

$$E[\hat{\boldsymbol{f}}\hat{\boldsymbol{f}}^{\mathrm{T}}] = [\boldsymbol{L}][\boldsymbol{h}][\boldsymbol{R}_f][\boldsymbol{h}]^{\mathrm{T}}[\boldsymbol{L}]^{\mathrm{T}} + [\boldsymbol{L}][\boldsymbol{R}_n][\boldsymbol{L}]^{\mathrm{T}}$$

代入式(5.4-9)得

$$\begin{aligned} e^2 = T_r\{[\boldsymbol{R}_f] - [\boldsymbol{R}_f][\boldsymbol{h}]^{\mathrm{T}}[\boldsymbol{L}]^{\mathrm{T}} - [\boldsymbol{L}][\boldsymbol{h}][\boldsymbol{R}_f] + \\ [\boldsymbol{L}][\boldsymbol{h}][\boldsymbol{R}_f][\boldsymbol{h}]^{\mathrm{T}}[\boldsymbol{L}]^{\mathrm{T}} + [\boldsymbol{L}][\boldsymbol{R}_n][\boldsymbol{L}]^{\mathrm{T}}\} \end{aligned}$$

式中第二项和第三项互为转置，所以它们的对角线元素相等，因此迹也相等，上式成为

$$e^2 = T_r\{[\boldsymbol{R}_f] - 2[\boldsymbol{R}_f][\boldsymbol{h}]^{\mathrm{T}}[\boldsymbol{L}]^{\mathrm{T}} + [\boldsymbol{L}][\boldsymbol{h}][\boldsymbol{R}_f][\boldsymbol{h}]^{\mathrm{T}}[\boldsymbol{L}]^{\mathrm{T}} + [\boldsymbol{L}][\boldsymbol{R}_n][\boldsymbol{L}]^{\mathrm{T}}\} \tag{5.4-12}$$

为了求矩阵$[\boldsymbol{L}]$使误差 e^2 最小，可将上式对矩阵$[\boldsymbol{L}]$求导，并令其等于 0。

标量对矩阵的导数可以这样来定义，即：标量 e^2 对矩阵$[\boldsymbol{L}]$的导数还是一个矩阵，它的第 i, j 个元素为 e^2 对矩阵$[\boldsymbol{L}]$中相应元素求导，即为

$$\frac{\partial e^2}{\partial[\boldsymbol{L}]} = \left[\frac{\partial e^2}{\partial L_{ij}}\right], \qquad 0 \leqslant i, j \leqslant PQ - 1$$

这里矩阵 $[\boldsymbol{L}]$ 为 $PQ \times PQ$ 阶矩阵。利用上面定义很容易证明下面两个性质

(1) 如果 $e^2 = T_r\{[\boldsymbol{A}][\boldsymbol{L}]^{\mathrm{T}}\}$，则

$$\frac{\partial e^2}{\partial[\boldsymbol{L}]} = [\boldsymbol{A}] \tag{5.4-13}$$

(2) 如果 $e^2 = T_r\{[\boldsymbol{L}][\boldsymbol{A}][\boldsymbol{L}]^{\mathrm{T}}\}$，则

$$\frac{\partial e^2}{\partial[\boldsymbol{L}]} = 2[\boldsymbol{L}][\boldsymbol{A}] \tag{5.4-14}$$

现将式(5.4-12)对矩阵$[\boldsymbol{L}]$求导，并令其为 0，得

$$\frac{\partial e^2}{\partial[\boldsymbol{L}]} = -2[\boldsymbol{R}_f][\boldsymbol{h}]^{\mathrm{T}} + 2[\boldsymbol{L}][\boldsymbol{h}][\boldsymbol{R}_f][\boldsymbol{h}]^{\mathrm{T}} + 2[\boldsymbol{L}][\boldsymbol{R}_n] = 0$$

由此可以解得使误差 e^2 最小的矩阵 $[\boldsymbol{L}]$

$$[\boldsymbol{L}] = [\boldsymbol{R}_f][\boldsymbol{h}]^{\mathrm{T}}\{[\boldsymbol{h}][\boldsymbol{R}_f][\boldsymbol{h}]^{\mathrm{T}} + [\boldsymbol{R}_n]\}^{-1} \tag{5.4-15}$$

下面设法将上式的解转换到频率域。根据前一节讨论可知，对于一个堆砌后的矩阵 $[\boldsymbol{A}]$ 来说，如果要使它经傅里叶变换以后为对角线矩阵，一定要是分块循环矩阵。就是说它的元素要满足

$$[\boldsymbol{A}]_{mn,kl} = a[((m-k))_P, ((n-l))_Q]$$

现在来看自相关矩阵 $\boldsymbol{R}_f$，如果 $f_e(m, n)$ 是一平稳随机过程，则

$$[\boldsymbol{R}_f]_{mn,kl} = E[f_e(m, n)f_e(k, l)] = E[f_e(0, 0)f_e(|m-k|, |n-l|)]$$

这说明 $[\boldsymbol{R}_f]$ 是一个分块的 Teoplitz 矩阵，不是分块循环矩阵。但是 Andrews 和 Hunt 认为可以用分块循环矩阵来近似这个分块 Toeplitz 矩阵，条件是

$$[\boldsymbol{R}_f]_{mn,kl} \approx 0, \quad \text{当}\ |m-k| + |n-l| > 20\ \text{或}\ 30$$

这个条件通常是满足的[1]。因此可以用分块循环矩阵 $[\boldsymbol{R}_f^c]$ 近似 $[\boldsymbol{R}_f]$，即

$$[\boldsymbol{R}_f^c] \approx [\boldsymbol{R}_f] \tag{5.4-16a}$$

同样有

$$[\boldsymbol{R}_n^c] \approx [\boldsymbol{R}_n] \tag{5.4-16b}$$

为方便起见，我们用 $PQ \times PQ$ 矩阵 $[\boldsymbol{T}]$ 表示直积矩阵 $[\boldsymbol{U}] \otimes [\boldsymbol{V}]$

$$[\boldsymbol{T}] = [\boldsymbol{U}] \otimes [\boldsymbol{V}] \tag{5.4-17}$$

所以有

$$[\boldsymbol{T}][\boldsymbol{T}]^* = [\boldsymbol{T}]^*[\boldsymbol{T}] = \boldsymbol{I}_{PQ} \tag{5.4-18}$$

由式(5.3-31)可以写出

$$[\boldsymbol{T}][\boldsymbol{h}][\boldsymbol{T}]^* = [\boldsymbol{\Lambda}_h]$$

$$= \begin{bmatrix} H(0, 0) & & & & & & \\ & H(0, 1) & & & & & \\ & & \ddots & & & & \\ & & & H(0, Q-1) & & & \\ & & & & H(1, 0) & & \\ & & & & & \ddots & \\ & & & & & & H(P-1, Q-1) \end{bmatrix} \tag{5.4-19}$$

这个式子对任何分块循环矩阵都成立，对角元素就是分块循环矩阵 $h(m, n)$ 的二维傅里叶变换。所以

$$[\boldsymbol{T}][\boldsymbol{R}_f'][\boldsymbol{T}]^* \approx [\boldsymbol{T}][\boldsymbol{R}_f][\boldsymbol{T}]^*$$

$$= \mathrm{diag}[S_f(0, 0)S_f(0, 1)\cdots S_f(1, 0)\cdots S_f(P-1, Q-1)] = [\boldsymbol{\Lambda}_f] \tag{5.4-20}$$

式中：

$$S_f(u, v) = \frac{1}{PQ}\sum_{m=0}^{P-1}\sum_{n=0}^{Q-1}R_f(m, n)\exp\left\{-\mathrm{j}2\pi\left(\frac{um}{P}+\frac{vn}{Q}\right)\right\} \tag{5.4-21}$$

同样有

$$[\boldsymbol{T}][\boldsymbol{R}_n'][\boldsymbol{T}]^* \approx [\boldsymbol{T}][\boldsymbol{R}_n][\boldsymbol{T}]^*$$

$$= \mathrm{diag}[S_n(0, 0)S_n(0, 1)\cdots S_n(1, 0)\cdots S_n(P-1, Q-1)] = [\boldsymbol{\Lambda}_n] \tag{5.4-22}$$

式中：

$$S_n(u, v) = \frac{1}{\sqrt{PQ}}\sum_{m=0}^{P-1}\sum_{n=0}^{Q-1}R_n(m, n)\exp\left\{-\mathrm{j}2\pi\left(\frac{um}{P}+\frac{vn}{Q}\right)\right\} \tag{5.4-23}$$

由式(5.4-15)有

$$[\boldsymbol{L}]\{[\boldsymbol{h}][R_f][\boldsymbol{h}]^{\mathrm{T}}+[\boldsymbol{R}_n]\} = [\boldsymbol{R}_f][\boldsymbol{h}]^{\mathrm{T}}$$

将等号两边的左右两端各乘以 $[\boldsymbol{T}]$ 和 $[\boldsymbol{T}]^*$，并利用式(5.4-18)，可以把上式写成

$$[\boldsymbol{T}][\boldsymbol{L}][\boldsymbol{T}]^*\{[\boldsymbol{T}][\boldsymbol{h}][\boldsymbol{T}]^*[\boldsymbol{T}][\boldsymbol{R}_f][\boldsymbol{T}]^*[\boldsymbol{T}][\boldsymbol{h}]^{\mathrm{T}}[\boldsymbol{T}]^*+[\boldsymbol{T}][\boldsymbol{R}_n][\boldsymbol{T}]^*\}$$
$$= [\boldsymbol{T}][\boldsymbol{R}_f][\boldsymbol{T}]^*[\boldsymbol{T}][\boldsymbol{h}]^{\mathrm{T}}[\boldsymbol{T}]^*$$

将式(5.4-19)，(5.4-20)和式(5.4-22)代入上式，并利用

$$[\boldsymbol{T}][\boldsymbol{h}]^{\mathrm{T}}[\boldsymbol{T}]^* = ([\boldsymbol{T}]^*[\boldsymbol{h}][\boldsymbol{T}])^{\mathrm{T}} = ([\boldsymbol{T}][\boldsymbol{h}][\boldsymbol{T}]^*)^{*\mathrm{T}} = [\boldsymbol{\Lambda}_h]^{*\mathrm{T}} = [\boldsymbol{\Lambda}_h]^* \tag{5.4-24}$$

可以得到

$$[\boldsymbol{T}][\boldsymbol{L}][\boldsymbol{T}]^*\{[\boldsymbol{\Lambda}_h][\boldsymbol{\Lambda}_f][\boldsymbol{\Lambda}_h]^*+[\boldsymbol{\Lambda}_n]\} \approx [\boldsymbol{\Lambda}_f][\boldsymbol{\Lambda}_h]^*$$

$$[\boldsymbol{T}][\boldsymbol{L}][\boldsymbol{T}]^* \approx \frac{[\boldsymbol{\Lambda}_f][\boldsymbol{\Lambda}_h]^*}{[\boldsymbol{\Lambda}_h][\boldsymbol{\Lambda}_f][\boldsymbol{\Lambda}_h]^*+[\boldsymbol{\Lambda}_n]} \tag{5.4-25}$$

可见 $[\boldsymbol{T}][\boldsymbol{L}][\boldsymbol{T}]^*$ 也近似为对角线矩阵，即 $[\boldsymbol{L}]$ 近似为分块循环矩阵。设

$$[\boldsymbol{T}][\boldsymbol{L}][\boldsymbol{T}]^* = [\boldsymbol{\Lambda}_L] = \mathrm{diag}[\mathscr{L}(0,0), \mathscr{L}(0,1), \cdots, \mathscr{L}(P-1, Q-1)]$$

式中：$\mathscr{L}(u, v)$为$L(m, n)$的二维傅里叶变换。若将式(5.4-25)的对角线元素写出来，则为

$$\mathscr{L}(u, v) \approx \frac{S_f(u, v)H^*(u, v)}{|H(u, v)|^2 S_f(u, v) + S_n(u, v)} \tag{5.4-26}$$

这是维纳滤波在离散情况下的近似解，它在形式上与连续情况下维纳滤波的公式(5.4-18)相同。

5.4.2 有约束条件复原

维纳滤波是将图像 f 看做一个平稳随机过程，以图像和噪声的相关矩阵为基础，在统计意义上求最小误差而得到的。因此它的结果在平均意义上最佳。现在讨论在确定的意义上是最佳的，即把解 f 看做一个确知解。为了求得合理解，需要加上一个约束条件。设 Q 为 f 的线性算子，即使

$$\|[\boldsymbol{Q}]\boldsymbol{f}\|^2 = \{[\boldsymbol{Q}]\boldsymbol{f}\}^{\mathrm{T}}[\boldsymbol{Q}]\boldsymbol{f} = \boldsymbol{f}^{\mathrm{T}}[\boldsymbol{Q}]^{\mathrm{T}}[\boldsymbol{Q}]\boldsymbol{f} \tag{5.4-27}$$

最小，同时满足图像的噪声能量为常数的约束条件，即

$$\|\boldsymbol{g} - [\boldsymbol{h}]\hat{\boldsymbol{f}}\|^2 = \{\boldsymbol{g} - [\boldsymbol{h}]\hat{\boldsymbol{f}}\}^{\mathrm{T}}\{\boldsymbol{g} - [\boldsymbol{h}]\hat{\boldsymbol{f}}\} = C$$

为了得到符合上述要求的解，利用拉格朗日乘子法得准则函数为

$$J[\hat{\boldsymbol{f}}] = \|[\boldsymbol{Q}]\hat{\boldsymbol{f}}\|^2 + \alpha[\|\boldsymbol{g} - [\boldsymbol{h}]\hat{\boldsymbol{f}}\|^2 - C] \tag{5.4-28}$$

式中：α，C 为常数。同样将上式对$\hat{\boldsymbol{f}}$求导并令其为0，就可得到要求的解。

$$\frac{\partial J}{\partial \hat{\boldsymbol{f}}} = 2[\boldsymbol{Q}]^{\mathrm{T}}[\boldsymbol{Q}]\hat{\boldsymbol{f}} - 2\alpha[\boldsymbol{h}]^{\mathrm{T}}\{\boldsymbol{g} - [\boldsymbol{h}]\hat{\boldsymbol{f}}\} = 0$$

由此解得

$$\boldsymbol{f} = \left\{[\boldsymbol{h}]^{\mathrm{T}}[\boldsymbol{h}] + \frac{1}{\alpha}[\boldsymbol{Q}]^{\mathrm{T}}[\boldsymbol{Q}]\right\}^{-1}[\boldsymbol{h}]^{\mathrm{T}}\boldsymbol{g} \tag{5.4-29}$$

与维纳滤波一样，为了简化计算，通常把它转到频率域上去。为此，矩阵$[\boldsymbol{Q}]$应该是分块循环矩阵。那么怎样来选择矩阵$[\boldsymbol{Q}]$呢？Phillips[2]提出过一种准则，即以图像的平滑度为基础。图像的平滑度常用图像的二阶导数，即拉普拉斯运算来表示。这种运算可以表示成算子 $q(m, n)$和图像 $f(m, n)$卷积。如5.3.2小节中所述，经周期延拓后可用循环卷积来表示，即

$$q(m, n) * f(m, n) = \sum_{k=0}^{P-1}\sum_{l=0}^{Q-1} q_e(m-k, n-l)f_e(k, l) \tag{5.4-30}$$

式中：P, Q 为延拓后的周期，q_e 和 f_e 为 q 和 f 延拓后的结果。同样，式(5.4－30)和向量表示为

$$\boldsymbol{q}=[\boldsymbol{Q}]\boldsymbol{f} \tag{5.4-31}$$

式中：$[\boldsymbol{Q}]$是一分块循环矩阵，矩阵元素为

$$[\boldsymbol{Q}]_{mn,kl}=q_e[((m-k))_P,((n-l))_Q] \tag{5.4-32}$$

在离散情况下，拉普拉斯运算定义为 3×3 大小，并有

$$q(0,1)=q(1,0)=q(1,2)=q(2,1)=1$$
$$q(1,1)=-4$$

其他的 $q(m,n)=0$，对应的模板如图 5.4 所示。

0	1	0
1	-4	1
0	1	0

图 5.4 拉普拉斯运算模板

由 5.3.2 小节中式(5.4－26)及式(5.4－27)可以把$\hat{f}$和 $\boldsymbol{g}$ 的傅里叶变换表示成

$$\hat{\boldsymbol{F}}=[\boldsymbol{T}]\hat{\boldsymbol{f}},\ \boldsymbol{G}=[\boldsymbol{T}]\boldsymbol{g} \tag{5.4-33}$$

式中：$[\boldsymbol{T}]=[\boldsymbol{U}]\otimes[\boldsymbol{V}]$。再利用对 $PQ\times PQ$ 分块循环矩阵$[\boldsymbol{h}]$和$[\boldsymbol{Q}]$的正交变换，得

$$\begin{aligned}[\boldsymbol{T}][\boldsymbol{h}][\boldsymbol{T}]^{*}=\operatorname{diag}[&H(0,0),H(0,1),\cdots,H(0,Q-1),H(1,0),\cdots,\\&H(P-1,Q-1)]=[\boldsymbol{A}_h]\end{aligned} \tag{5.4-34}$$

$$\begin{aligned}[\boldsymbol{T}][\boldsymbol{Q}][\boldsymbol{T}]^{*}=\operatorname{diag}[&Q(0,0),Q(0,1),\cdots,Q(0,Q-1),Q(1,0),\cdots,\\&Q(P-1,Q-1)]=[\boldsymbol{A}_q]\end{aligned} \tag{5.4-35}$$

式中：

$$Q(u,v)=\sum_{m=0}^{P-1}\sum_{n=0}^{Q-1}q_e(m,n)\exp\left\{-j\left(\frac{2\pi}{P}mu+\frac{2\pi}{Q}nv\right)\right\}$$
$$u=0,1,\cdots,P-1;\ v=0,1,\cdots,Q-1 \tag{5.4-36}$$

式(5.4－29)可以写成

$$\left\{[\boldsymbol{h}]^{\mathrm{T}}[\boldsymbol{h}]+\frac{1}{\alpha}[\boldsymbol{Q}]^{\mathrm{T}}[\boldsymbol{Q}]\right\}\hat{\boldsymbol{f}}=[\boldsymbol{h}]^{\mathrm{T}}\boldsymbol{g}$$

利用 $[\boldsymbol{T}]^{*}[\boldsymbol{T}]=\boldsymbol{I}$，并将上式左乘$[\boldsymbol{T}]$，得

$$\begin{aligned}&[\boldsymbol{T}]\left\{[\boldsymbol{h}]^{\mathrm{T}}[\boldsymbol{T}]^{*}[\boldsymbol{T}][\boldsymbol{h}][\boldsymbol{T}]^{*}[\boldsymbol{T}]+\frac{1}{\alpha}[\boldsymbol{Q}]^{\mathrm{T}}[\boldsymbol{T}]^{*}[\boldsymbol{T}][\boldsymbol{Q}][\boldsymbol{T}]^{*}[\boldsymbol{T}]\right\}\hat{\boldsymbol{f}}\\&=[\boldsymbol{T}][\boldsymbol{h}]^{\mathrm{T}}[\boldsymbol{T}]^{*}[\boldsymbol{T}]\boldsymbol{g}\end{aligned}$$

将式(5.4－24)和式(5.4－33)至式(5.4－35)代入上式，可得

$$\left\{[\boldsymbol{A}]_h^*[\boldsymbol{A}_h]+\frac{1}{\alpha}[\boldsymbol{A}]_q^*[\boldsymbol{A}_q]\right\}\hat{\boldsymbol{F}}=[\boldsymbol{A}_h]^*\boldsymbol{G} \tag{5.4-37}$$

若按元素则可以写成

$$\hat{F}(u,v)=\frac{H^*(u,v)G(u,v)}{|H(u,v)|^2+\frac{1}{\alpha}|Q(u,v)|^2} \tag{5.4-38}$$

式中：α是一个未定常数。将上式与(5.4-26)式比较，可以看出α与噪声有关。设$\beta=\frac{1}{\alpha}$，β为待定的参量。为了得到满足约束条件 $\|\boldsymbol{g}-[\boldsymbol{h}]\hat{\boldsymbol{f}}\|^2=\|\boldsymbol{n}\|^2$ 的解，应调整参量β直至这一条件得到满足。β的估算可用下面的迭代过程进行。

定义一剩余向量$\boldsymbol{r}$为

$$\boldsymbol{r}=\boldsymbol{g}-[\boldsymbol{h}]\hat{\boldsymbol{f}}=\boldsymbol{g}-[\boldsymbol{h}]\{[\boldsymbol{h}]^{\mathrm{T}}[\boldsymbol{h}]+\beta[\boldsymbol{Q}]^{\mathrm{T}}[\boldsymbol{Q}]\}^{-1}[\boldsymbol{h}]^{\mathrm{T}}\boldsymbol{g} \tag{5.4-39}$$

所以$\boldsymbol{r}$是β的函数。我们的目的就是要确定β的值，使

$$\Phi(\beta)=\|\boldsymbol{r}\|^2=\|\boldsymbol{n}\|^2-\theta \tag{5.4-40}$$

式中：θ是准确度因子。因此，需要研究β和$\boldsymbol{r}$的变化关系[3]。

当β增加时，由式(5.4-38)可见$F(u,v)$下降。由巴塞伐尔(Parseval)定理

$$\sum_u\sum_v|F(u,v)|^2=\sum_m\sum_n|f(m,n)|^2$$

所以$|\boldsymbol{f}|$也减小。当$|\boldsymbol{g}|\gg|\boldsymbol{r}|$时，由式(5.4-39)可以看出$\boldsymbol{g}$和$[\boldsymbol{h}]\hat{\boldsymbol{f}}$方向几乎相反，长度几乎相等。$[\boldsymbol{h}]$由点扩展函数形成，所以$[\boldsymbol{h}]$元素都是正的。这样，一般说来，$|\boldsymbol{f}|$减小，$[\boldsymbol{h}]\boldsymbol{f}$也减小，$\boldsymbol{r}$增加。所以总的来说，$\beta$增加，$|\boldsymbol{r}|$也增加。因为$\Phi(\beta)$是单调增函数，可以用试探性的迭代方法来计算$\beta$。一个简单的方法是，先假定$\beta$的初始值，由式(5.4-38)求出解为$\hat{\boldsymbol{f}}_0$，然后求出$\|\boldsymbol{r}\|^2$，若$\Phi(\beta)<\|\boldsymbol{n}\|^2-\theta$，则增加$\beta$后重新计算，如果$\Phi(\beta)>\|\boldsymbol{n}\|^2+\theta$，则减少$\beta$后重新计算，直到收敛为止，即得$\beta$的准确值。

本节介绍了无"约束条件"的反滤波及维纳滤波的复原方法，还介绍了有约束的最小平方滤波的复原方法。其他约束条件下的复原问题未予讨论。这些方法都是线性代数的复原方法。和代数方法相对应的还有用时间序列的卡尔曼(Kalman)滤波复原等，这里不一一介绍了。

5.5 运动模糊图像的复原

在照相的曝光期间，如果物体和相机有相对运动，则反映在底片上的图像也有

明显移动，这样形成的图像称为运动模糊图像。这种现象在高速相对运动情况下拍摄照片，而底片的感光灵敏度不够高，曝光时间又太长时常会发生。这一节讨论匀速直线运动情况下拍摄的模糊图像的复原。

设运动方向为 x 轴方向，则图像函数可以写成 $f(x-x_0(t),\ y)$，这里 $x_0(t)$是沿 x 的正方向运动的距离，见图 5.5。设曝光时间为 $0 \leqslant t \leqslant T$，在此期间相片上的移动距离是 a，所以 $x_0(t) = at/T$。形成的模糊图像为

$$g(x,\ y) = \alpha\int_0^T f\left(x - \frac{at}{T},\ y\right)\mathrm{d}t \tag{5.5-1}$$

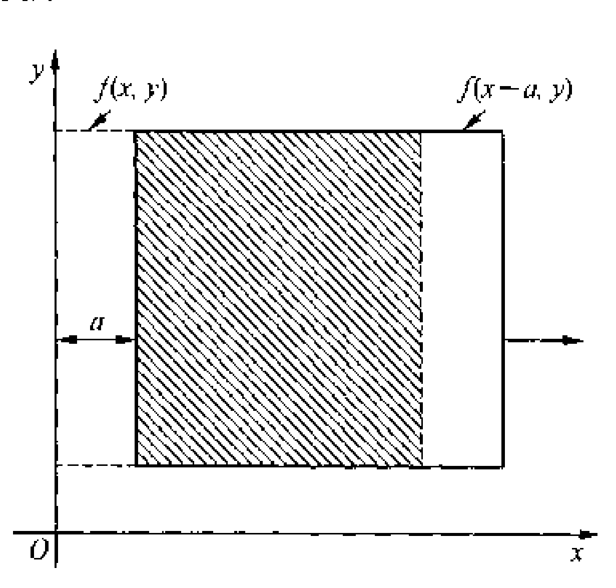


图 5.5　图像的运动模糊

式中 α 是和照片感光灵敏度有关的系数。现在已知 $g(x,\ y)$和 a，求 $f(x,\ y)$。这就是最简单的运动模糊图像的复原问题。因为现在假设模糊只是在 x 方向，和 y 方向无关，所以可以一行一行地复原。在某一行 $y = y_0$ 时，把 $g(x,\ y_0)$和$f(x,\ y_0)$简写成 $g(x)$和 $f(x)$。这时式(5.5-1) 可以改写为：

$$g(x) = \alpha\int_0^T f\left(x - \frac{at}{T}\right)\mathrm{d}t \tag{5.5-2}$$

对上式作变量置换，令 $x' = x - \dfrac{at}{T}$，则

$$g(x) = -\frac{T\alpha}{a}\int_x^{x-a} f(x')\mathrm{d}x'$$

对上式求导得

$$\frac{\mathrm{d}g(x)}{\mathrm{d}x} = -\frac{T\alpha}{a}f(x')\Big|_x^{x-a} = \frac{T\alpha}{a}[f(x) - f(x-a)]$$

即

$$g'(x) = \frac{1}{\beta}[f(x) - f(x-a)] \tag{5.5-3}$$

上式可改写成

$$f(x) = \beta g'(x) + f(x-a) \tag{5.5-4}$$

式中：$\beta = a/T\alpha$。上式说明 $f(x)$ 有递推性质。模糊图像 $g(x)$ 的导数是可以求得的，所以已知 $f(x-a)$ 就可以推得 $f(x)$。因此实际上只要求出总长度为 a 的区间上的图像值，就可以推得整个图像了。

设研究的区域为 $0 \leqslant x \leqslant L$，$L = Ka$，$K$ 为正整数。即将这一区域分成 K 段长度为 a 的子区域，则第 m 段区域的 x 值为：

$$x = z + ma,\ m = 0,\ 1,\ \cdots,\ K-1 \tag{5.5-5}$$

式中：z 为区间 $[0,\ a)$ 中的值，即 $0 \leqslant z < a$。将上式代入式 (5.5-4) 得

$$f(z+ma) = \beta g'(z+ma) + f[z+(m-1)a] \tag{5.5-6}$$

设 $\varphi(z)$ 为图像在曝光时间 T 内，在 $0 \sim a$ 范围的移动部分，即在 $(-a,\ 0)$ 上图像值有

$$\varphi(z) = f(z-a),\ 0 \leqslant z < a \tag{5.5-7}$$

这样式(5.5-6)就可以用以下的递推关系求得。当 $m = 0$ 时，

$$f(z) = \beta g'(z) + \varphi(z)$$

$m = 1$ 时，

$$f(z+a) = \beta g'(z+a) + f(z) = \beta[g'(z+a) + g'(z)] + \varphi(z)$$

$m = 2$ 时，

$$\begin{aligned} f(z+2a) &= \beta g'(z+2a) + f(z+a) \\ &= \beta[g'(z+2a) + g'(z+a) + g'(z)] + \varphi(z) \\ &\vdots \end{aligned}$$

由此可以得出它的一般表达式：

$$f(z+ma) = \beta\sum_{k=0}^{m} g'(z+ka) + \varphi(z),\ 0 \leqslant z < a \tag{5.5-8}$$

式中不论 m 为何值，$\varphi(z)$ 总是存在的。因此它在图像中是一个周期函数。当 $ma \leqslant x < (m+1)a$ 时，

$$\begin{aligned} f(x) &= \beta\sum_{k=0}^{m} g'(x-ma+ka) + \varphi(x-ma) \\ &= \beta\sum_{k=0}^{m} g'(x-ka) + \varphi(x-ma),\ 0 \leqslant x < L \end{aligned} \tag{5.5-9}$$

上面几式中，g' 可以由模糊的图像求得，但 $\varphi(z)$ 是一未知函数。对于任何一个 $\varphi(z)$ 都有一个满足式(5.5-1)的解。所以只有在确定了某一长度为 a 的区间的图像以后才能确定 $\varphi(z)$[4]。下面介绍一种粗糙的近似方法。

由式(5.5-8)有

$$\varphi(z)=f(z+ma)-\beta\sum_{k=0}^{m}g'(z+ka)$$

式中 $m=0, 1, \cdots, K-1$，共有 K 项。将上式以 K 项相加得

$$K\varphi(z)=\sum_{m=0}^{K-1}f(z+ma)-\beta\sum_{m=0}^{K-1}\sum_{k=0}^{m}g'(z+ka)$$

$$\varphi(z)=\frac{1}{K}\sum_{m=0}^{K-1}f(z+ma)-\frac{\beta}{K}\sum_{m=0}^{K-1}\sum_{k=0}^{m}g'(z+ka) \qquad (5.5-10)$$

上式中第一项当 K 值较大时趋向于 f 的平均值，因此可将它近似地看做一个常数，所以设

$$\frac{1}{K}\sum_{m=0}^{K-1}f(z+ma)=A$$

则式(5.5-10)可以写成

$$\begin{aligned}\varphi(z)&\approx A-\frac{\beta}{K}\sum_{m=0}^{K-1}\sum_{k=0}^{m}g'(z+ka)\\&=A-\frac{\beta}{K}\{Kg(z)+(K-1)g'(z+a)+\cdots+g'[z+(K-1)a]\}\end{aligned} \qquad (5.5-11)$$

这里 A 是一个未知数，要用试验的方法确定。将上式代入式(5.5-8)，得

$$f(z+ma)\approx A+\beta\sum_{k=0}^{m}g'(z+ka)-\frac{\beta}{K}\sum_{m=0}^{K-1}\sum_{k=0}^{m}g'(z+ka)$$

最后可得

$$f(x)\approx A+\beta\sum_{k=0}^{m}g'(x-ka)-\frac{\beta}{K}\sum_{m=0}^{K-1}\sum_{k=0}^{m}g'(x-ka) \qquad (5.5-12)$$

以上讨论的运动模糊图像的复原只考虑了 x 方向上的匀速直线运动。但利用上述原理是不难得到一个同时考虑两个方向上匀速直线运动的去模糊表达式的。

下面将讨论把运动模糊和前面提到的某些复原方法联系起来，以得到另一类运动模糊图像的复原方法。

图像退化的数学模型可以写成

$$g(x, y)=h(x, y)*f(x, y)+n(x, y)$$

其傅里叶变换表示为

$$G(u, v) = H(u, v)F(u, v) + N(u, v)$$

式中：$n(x, y)$和$N(u, v)$是噪声项，是由成像系统本身引起的，与运动无关。

现在将式(5.5-2)扩展为二维函数的积分，即运动方向沿向量$\alpha i + \beta j = v$的方向，所以可以将运动模糊图像表示成积分

$$g(x, y) = \alpha\int_0^T f(x-\alpha t, y-\beta t)\mathrm{d}t + n(x, y) \tag{5.5-13}$$

这是空间域中的卷积方程，从这里求点扩展函数比较麻烦，为此将它变换到频域。重写连续情况下逆傅里叶变换式

$$f(x, y) = \iint_{-\infty}^{+\infty} F(u, v)\exp\{\mathrm{j}2\pi(ux+vy)\}\mathrm{d}u\mathrm{d}v$$

则式(5.5-13)中：

$$\begin{aligned}
&\int_0^T f(x-\alpha t, y-\beta t)\mathrm{d}t \\
&= \int_0^T\left[\iint_{-\infty}^{+\infty} F(u, v)\exp\{\mathrm{j}2\pi[u(x-\alpha t)+v(y-\beta t)]\}\mathrm{d}u\mathrm{d}v\right]\mathrm{d}t \\
&= \iint_{-\infty}^{+\infty} F(u, v)\exp\{\mathrm{j}2\pi(ux+vy)\}\left\{\int_0^T \exp[-\mathrm{j}2\pi(\alpha u+\beta v)t]\mathrm{d}t\right\}\mathrm{d}u\mathrm{d}v \\
&= \iint_{-\infty}^{+\infty} F(u, v)\frac{\mathrm{e}^{-\mathrm{j}\pi(\alpha u+\beta v)}\sin[\pi(\alpha u+\beta v)T]}{\pi(\alpha u+\beta v)}\mathrm{e}^{\mathrm{j}2\pi(ux+vy)}\mathrm{d}u\mathrm{d}v
\end{aligned}$$

所以上式中的被积函数即为$\int_0^T f(x-\alpha t, y-\beta t)\mathrm{d}t$的傅里叶变换。则式(5.5-13)的频域表示为

$$G(u, v) = F(u, v)H(u, v) + N(u, v)$$

式中：

$$H(u, v) = \alpha\mathrm{e}^{-\mathrm{j}\pi(\alpha u+\beta v)}\sin[\pi(\alpha u+\beta v)T]/[\pi(\alpha u+\beta v)] \tag{5.5-14}$$

有了$H(u, v)$就可以用维纳滤波或反滤波等方法来复原匀速直线运动形成的模糊图像。由式(5.5-14)可以看出，$H(u, v)$是由一种常见的特殊函数$\sin x/x$再加上相移组成的，从这一特点可以看出它包含着运动方向和距离的信息。

以上介绍的两种运动模糊图像复原，效果都不太满意，需要进一步发展。

5.6 图像重建

根据物体横剖面的一组投影数据，经过计算机处理后得到物体该横剖面的图像，这种由数据到图像的处理技术，称之为图像重建。图像重建技术的历史比较悠久。它的理论起源于1917年奥地利数学家雷顿(Radon)发表论文，从数学上证明了二维或三维物体能够通过其无限个投影来确定，但限于当时的技术条件而未能实现。直至六十年代初，随着技术的进步，特别是计算机科学的发展，图像重建问题才重新引起了人们的兴趣，不少学者进行了卓有成效的研究。英国EMI公司中央研究所工程师豪斯菲尔德(Housefield)在这一领域取得了很大成功，经过四年的努力，在1972年制成头颅用的电子计算机X断层摄影装置。并在1974年5月蒙特利尔召开的第一次国际专题讨论会上正式被命名为电子计算机断层摄影术(Computed Tomography, CT)。1975年EMI公司又成功地研究出全身用的CT装置并得到人体各个部位的鲜明清晰的断层图像。1979年，这项技术获得了诺贝尔奖金。

图像重建在许多领域有广泛的应用，它极大地增强了人类观察物体内部结构的能力，特别是在医学领域中，它已经被作为一种获取人体内部结构信息的有效手段，完全改变了放射诊断医学的面貌。

图像重建可以看做一类特殊的图像复原技术。如果将投影数据看成一种退化过程，则重建就是一种图像复原过程。根据投影重建的原理和方法，投影重建可以有不同的类型，这里我们主要讨论透射投影成像，即断层成像术。

图像重建方法大致分为两类。一类是直接法，典型的是代数重建法，这是第一代CT所采用的算法。另一类是间接法，典型的有傅里叶变换法和滤波—反投影法(Filtered-Backprojection Algorithm)。傅里叶变换法计算量比较大，现已不采用。目前CT所采用的算法均为滤波—反投影法，因为它有很高的精度，且能够快速实现。它主要采用对各投影向量先进行高通滤波，然后反投影回去，使与投影值有关的投影路径上的像素值的修改平均分配到各有关像素中去，也有按照像素对投影贡献大小来分配的，所有投影经过高通滤波后反投影回去，则构成了重建图像 $f(x, y)$。本章将逐一介绍这些典型的重建算法。

5.6.1 断层摄影图像的获取

断层摄影图像的获取通常采用发射接收方式，即以平行的X射线照射从各个不同的方向照射目标物体，并记录每一方向的透射场，如图5.6所示。X射线

源产生平行的X射线对目标物体进行照射，设其入射光强度为 I_0，接收器阵列得到透射光强度为 $I(t,\ \theta_1)$。依次类推，X源和X射线接收器沿中心转一个角度到 θ_2，又可以得到一组投影数据 $I(t,\ \theta_2)$，从0°到180°每隔一定间隔改变 θ，则得一组投影向量。

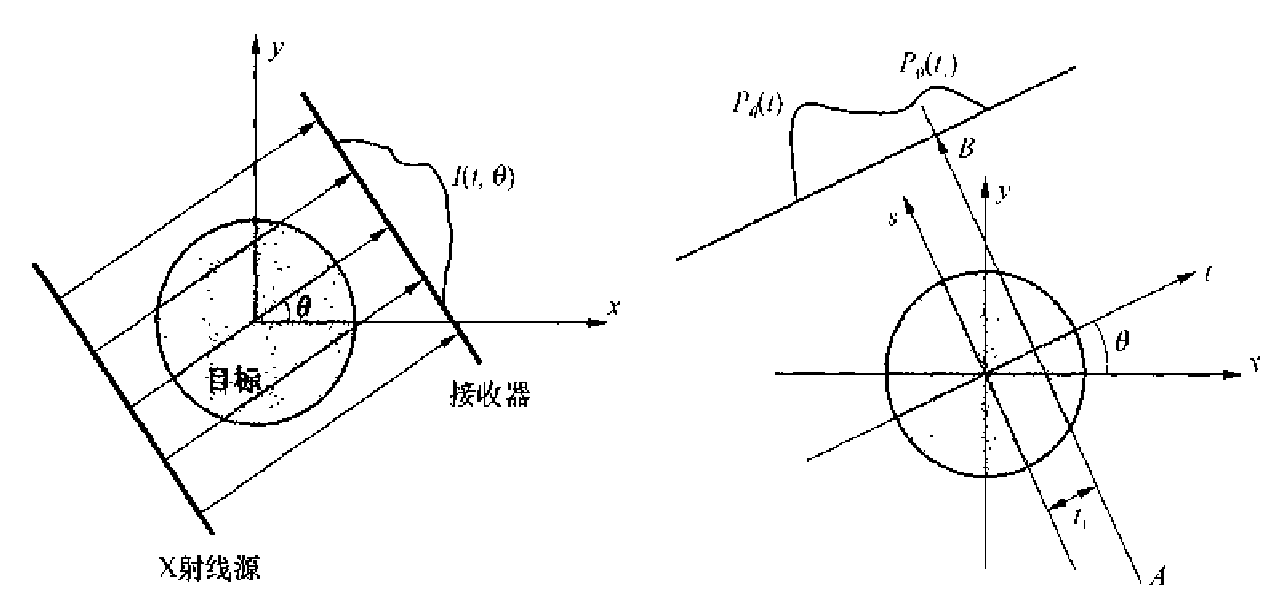


图5.6　断层摄影图像获取示意图　　　　图5.7　图像在 θ 角的投影 $P_\theta(t)$ 示意图

图像的投影在数学上可以描述如下。如图5.7所示，令 $f(x,\ y)$ 表示图像函数，穿过 $f(x,\ y)$ 的一条线称为射线。$f(x,\ y)$ 沿某一射线的积分称为射线积分，而射线积分的集合则组成投影。

若从坐标原点向射线作一垂线，以此垂线作为新坐标的一个轴 t，并构成新坐标系 $(t,\ s)$，可以看出 $(t,\ s)$ 坐标系仅是 $(x,\ y)$ 坐标系旋转 θ 角的结果，二者存在下列变换关系：

$$\begin{bmatrix} t \\ s \end{bmatrix} = \begin{bmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} \tag{5.6-1}$$

因而射线积分可以表达为：

$$\boldsymbol{P}_\theta(t_1) = \int_{AB} f(x,\ y)\mathrm{d}x\mathrm{d}y = \int_{-\infty}^{\infty} f(t_1,\ s)\mathrm{d}s \tag{5.6-2}$$

沿一组垂直于 t 的射线积分，即形成了 $f(x,\ y)$ 在 θ 角位置的平行投影，二维函数 $\boldsymbol{P}_\theta(t)$ 也叫 $f(x,\ y)$ 的雷顿(Radon)变换，相应的重建过程也称Radon反变换。

断层成像技术就是从不同射线角度 θ，不同检测器的位置的许多投影值 $\boldsymbol{P}_\theta(t)$ 重建原始图像 $f(x,\ y)$ 的过程，$f(x,\ y)$ 反应了 $(x,\ y)$ 处的密度。

在实际过程中，角度不可能无限多，检测元素也是有限的，设从0到180度每隔3度求一次投影，则共有60个投影向量，记作 P 个投影向量，每个向量 N 维，则

共有 PN 个投影值，需要求解 $N \times N$ 个 $f(m, n)$。

和图像复原一样，重建也会碰到病态问题，如 $N = 2$，$P = 2$，则不能从 4 个投影值唯一地解出 A，B，C 和 D 4 个像素值。解决办法是增加投影向量 P 的数量，可消去不定性；或者根据先验知识加上约束条件；或者降低分辨率，这样要求的点数就少了，但结果是重建的图像分辨率也相应地降低。

5.6.2 代数重建法

1. 算法原理

代数重建法(Algebraic Reconstruction Technique，ART)是在离散域中重建图像。

设要重建的图像 $[f(m, n)]_{N\times N}$ 位于一正方形网格中，每个小方格表示一像素。将 $[f(m, n)]_{N\times N}$ 按行堆叠成一个 $N^2 \times 1$ 的列向量 $\boldsymbol{f}$。在角度 θ_k 方向的投影向量为 $\boldsymbol{P}_k$(设为 N 维，即有 N 个检测器)

$$\boldsymbol{P}_k = (P_{k1}, \cdots, P_{kN})^{\mathrm{T}} \tag{5.6-3}$$

若共有 N_θ 个投影向量，把这些 $\boldsymbol{P}_k(k = 1, \cdots, N_\theta)$ 堆叠起来，可构成维数为 $N_\theta \times N$ 的列向量 $\boldsymbol{P}$，即

$$\boldsymbol{P} = (\boldsymbol{P}_1^{\mathrm{T}}, \boldsymbol{P}_2^{\mathrm{T}}, \cdots, \boldsymbol{P}_{N_\theta}^{\mathrm{T}})^{\mathrm{T}} \tag{5.6-4}$$

在代数重建方法中，设射线具有一定的宽度，通常射线宽度与小方格宽度相等。显然，对第 k 个投影方向的某个投影值的大小起作用的截面上的像素与射线经过的几何位置有关。

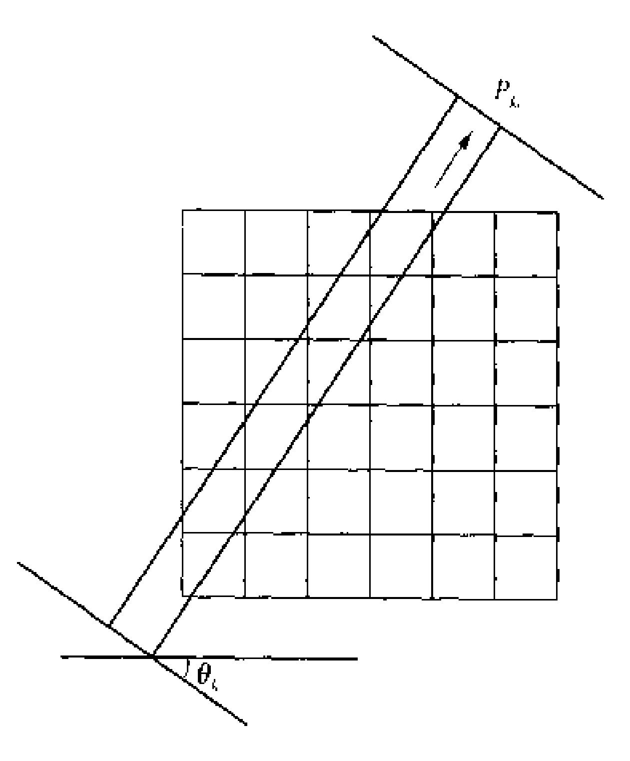


图 5.8 代数重建的各种参数

如图 5.8 所示，在射线束扫过的位置上的像素对 $P_{ki}(i = 1, \cdots, N)$ 投影值都有影响。而射线束外对 P_{ki} 无影响，其影响的大小可以用 2 种方法来表示。

第一种：以射线束扫过该像素的面积与像素面积之比值来作为加权因子。

第二种：不管加权因子，射线扫中像素中心点为 1，认为此像素对 P_{ki} 有影响，反之为 0。

若用 b_{ij} 来表示像素 j 对第 i 个投

影值的影响，这里

$$j = mN + n,\ m,\ n = 0,\ \cdots,\ N-1;\ j = 0,\ \cdots,\ N^2-1$$

$$i = kN + l,\ k = 1,\ \cdots,\ N_\theta;\ l = 1,\ \cdots,\ N$$

则按第一种表示法 $b_{i,j} = 0$(线外)，或者 $0 < b_{ij} \leqslant 1$；按第二种表示法，射线扫中像素中心时 $b_{i,j} = 1$，未被扫中 $b_{i,j} = 0$。在离散情况下，射线的积分用累加和表示，则第 i 条射线得到的投影值为

$$P_i = \sum_{j=1}^{N^2} b_{ij} f_j \tag{5.6-5}$$

由于只有那些与射线相交的像素才对累加和作出贡献，因此大多数权系数 $b_{i,j}$ 都为零。若用矩阵形式表示，则投影值与像素衰减系数间的关系为

$$\boldsymbol{P} = [\boldsymbol{B}]f \tag{5.6-6}$$

式中：$\boldsymbol{P}$ 为 $N_\theta \times N = M$ 维列向量，$\boldsymbol{f}$ 为 N^2 维列向量，$[\boldsymbol{B}]$是个 $M \times N^2$ 维的巨大矩阵，也是个稀疏矩阵。若 $B_{i,j} = (0/1)$ 表示更方便，但用面积比更合理。由 $\boldsymbol{P} = [\boldsymbol{B}]f$ 可知投影的过程是已知 $\boldsymbol{f}$ 求 $\boldsymbol{P}$ 的过程，即求加权和的过程。而重建截面的过程是个求解线性方程组的过程。但是，这一方程组十分庞大，直接求解是困难的。而且，即使 $N_\theta = N$，$[\boldsymbol{B}]$成为方阵，也不一定有唯一解，即病态问题，这时可用先验知识，如截面的吸收在邻点较均匀等，加约束条件来正则化。按常规的方法不一定有唯一解，但可以在某种意义下求得最优解。下面介绍求解的近似算法。

2. 投影迭代法

投影迭代法的迭代公式为

$$\boldsymbol{f}^{(k)} = \boldsymbol{f}^{(k-1)} + \frac{\hat{\boldsymbol{P}}_j - \boldsymbol{b}_j \boldsymbol{f}^{(k-1)}}{\|\boldsymbol{b}_j\|^2} \boldsymbol{b}_j = \boldsymbol{f}^{(k-1)} + \Delta \boldsymbol{f}^{(k)} \tag{5.6-7}$$

式中：$f = (f_1,\ f_2,\ \cdots,\ f_{N^2})$；$\boldsymbol{f}^{(k)}$ 表示 f 的第 k 次迭代值，$\hat{P}_j$ 为沿第 j 条射线实际测得的投影值，$\boldsymbol{b}_j = (b_{j1},\ b_{j2},\ \cdots,\ b_{jN^2})$。迭代求解的思路是先对 N^2 个 $f(m,\ n)$ 赋初始值 $\boldsymbol{f}^{(0)}$，然后求一条射线的投影值 $P_j^{(0)}$，可求得观测投影向量$\hat{\boldsymbol{P}}_j$ 与估计的向量$\boldsymbol{P}_j^{(0)}$之差值，据此求得 $\Delta f^{(0)}$ 来修正投影路径上有关像素的值，得 $\boldsymbol{f}^{(1)}$。对所有射线重复以上过程($j = 1,\ \cdots,\ M$)，即完成第一次迭代。重复以上过程直到收敛，即修正量 $\Delta \boldsymbol{f}^{(k)}$ 小到允许的程度。

综合上面的思想，可以归纳出迭代步骤如下：

(1) 任意设定初始截面值 $\boldsymbol{f}^{(0)}$，常设为 0，也可以根据先验知识来设定，设 $k = 0$。

(2) 通过 $\boldsymbol{P}^{(k)} = [\boldsymbol{B}]\boldsymbol{f}^{(k)}$，求出 $\boldsymbol{P}^{(k)}$。

(3) 由观测向量 $\hat{\boldsymbol{P}}$ 和 $\boldsymbol{P}^{(k)}$ 可以求得 $\Delta\boldsymbol{P}^{(k)} = \hat{\boldsymbol{P}} - \boldsymbol{P}^{(k)}$；

(4) 用式(5.6-7)对 $f^{(k)}$ 进行修正得 $f^{(k+1)}$(对所有射线进行,即 $j = 1, \cdots, M$)。

(5) $k = k+1$,返回(2)直到收敛。

5.6.3 傅里叶变换法

对大多数重建算法来说,傅里叶变换方法是重建技术的基础。而傅里叶变换重建是以投影层析定理为基础的。这一定理可以描述如下:图像 $f(x, y)$ 在 θ 角度的平行投影 $\boldsymbol{P}_\theta$ 的傅里叶变换 $\boldsymbol{S}_\theta(\omega)$ 等于 $f(x, y)$ 的傅里叶变换 $F(u, v)$ 的一个层面,且与轴 ω_1 的夹角为 θ_k,即 $F(\omega, \theta)$。换句话说,平行投影 $\boldsymbol{P}_\theta$ 的傅里叶变换提供了 $f(x, y)$ 的傅里叶变换 $F(u, v)$ 沿直线 AB 的值,如图 5.9 所示。

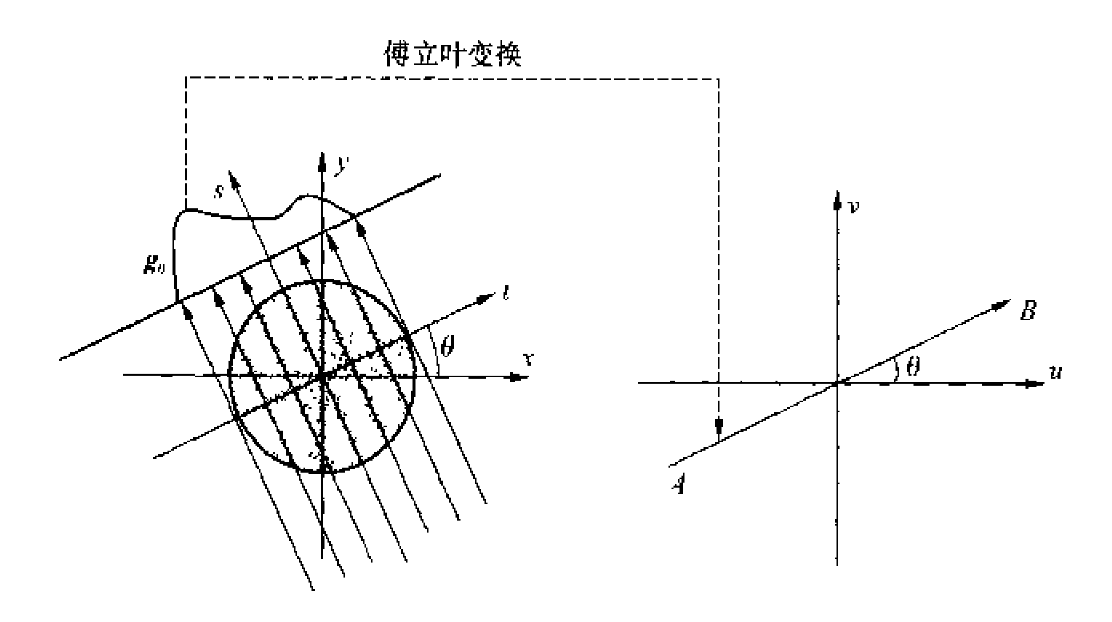


图 5.9 Fourier 层析理示意图

为了证明投影层析定理,令 $F(u, v)$ 为图像 $f(x, y)$ 的傅里叶变换,即

$$F(u, v) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} f(x, y)\exp[-j2\pi(\omega_1 x + \omega_2 y)]\mathrm{d}x\mathrm{d}y \tag{5.6-8}$$

设 $\boldsymbol{S}_\theta(\omega)$ 是投影 $\boldsymbol{P}_\theta$ 的傅里叶变换,

$$\boldsymbol{S}_\theta(\omega) = \int_{-\infty}^{\infty} \boldsymbol{P}_\theta(t)\exp(-j\omega t)\mathrm{d}t \tag{5.6-9}$$

现在我们观察 (u, v) 平面中直线 $v = 0$ 上的值,即

$$\begin{aligned} F(u, 0) &= \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} f(x, y)\exp[-j2\pi ux]\mathrm{d}x\mathrm{d}y \\ &= \int_{-\infty}^{\infty}\left[\int_{-\infty}^{\infty} f(x, y)\mathrm{d}y\right]\exp[-j2\pi ux]\mathrm{d}x \end{aligned}$$

$$= \int_{-\infty}^{\infty} P_0(t)\exp(-j\omega t)dt == S_0(\omega) \quad (5.6-10)$$

上式结果表明，对垂直于 x 轴所测得的投影 $P_0(t)$ 所作的傅里叶变换等于轴 u 上物体函数的傅里叶变换，推广到 θ 不等于 0 的情况，可以导出类似结果。

设 $F(\omega, \theta)$ 是 $F(u, v)$ 沿 θ 方向的值，$S_\theta(\omega)$ 是投影 $P_\theta(t)$ 的傅里叶变换，则有

$$F(\omega, \theta) = S_\theta(\omega) \quad (5.6-11)$$

首先把 $x-y$ 坐标轴旋转一个角度 θ，形成 $t-s$ 轴，(t, s) 与 (x, y) 坐标的关系如前面所述为：

$$\begin{bmatrix} t \\ s \end{bmatrix} = \begin{bmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix}$$

以 t，s 来重写式(5.6-9)，

$$S_\theta(\omega) = \int_{-\infty}^{\infty} P_\theta(t)e^{-j\omega t}dt = \int_{-\infty}^{\infty}\left[\int_{-\infty}^{\infty} f(t, s)ds\right]e^{-j\omega t}dt \quad (5.6-12)$$

变换到 (x, y) 坐标，得到：

$$\begin{aligned} S_\theta(\omega) &= \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} f(x, y)\exp[-j\omega(x\cos\theta + y\sin\theta)]dxdy \\ &= F(\omega\cos\theta, \omega\sin\theta) \end{aligned} \quad (5.6-13)$$

很明显，在时间域内旋转 θ 角度对应于在频率域内旋转 θ 角度，因此，推广后的一般关系式，以极坐标 (ω, θ) 来写出，就变为：

$$F(\omega, \theta) = S_\theta(\omega)$$

由此可见，如果逐一对 θ_1，θ_2，…，θ_k 各个方向上的投影进行傅里叶变换，我们就能获得 $F(u, v)$ 在这些角度的径线上的值，如图 5.10 所示。

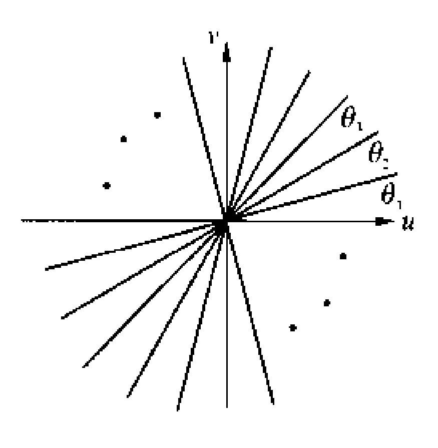


图 5.10 $F(u, v)$ 在不同角度径线上的取值

因此，当投影数量足够多(k 足够大)时，就能求得 $F(u, v)$ 在 (u, v) 平面上所有点的值，从而利用傅里叶反变换就可以重建图像 $f(x, y)$。在离散情况下，我们只能取得 $F(u, v)$ 在有限数目径线上有限个值。在进行傅里叶反变换之前，要根据有限的 $F(\omega, \theta)$ 离散值通过插值以求得 $F(u, v)$ 在矩形网格点上的值。由于 $F(\omega, \theta)$ 的数值点离开原点越远就越稀疏，插值的误差也越大。这意味着重建图像的高频

分量的误差要比低频分量大。

5.6.4 滤波-反投影法

目前CT所采用的算法均为滤波—反投影法，因为它有很高的精度，且能够快速实现。滤波-反投影法又称卷积-反投影法。本节主要讨论平行投影数据的算法。

如前所述，若(ω, θ)表示(u, v)平面上的极坐标，利用极坐标与平面坐标的关系，可将傅里叶反变换写成如下形式：

$$f(x, y) = \int_0^{2\pi}\int_0^{\infty} F(\omega, \theta)\exp[j\omega(x\cos\theta + y\sin\theta)]\omega d\omega d\theta$$

$$= \int_0^{\pi}\int_0^{\infty} F(\omega, \theta)\exp[j\omega(x\cos\theta + y\sin\theta)]\omega d\omega d\theta$$

$$+ \int_0^{\pi}\int_0^{\infty} F(\omega, \theta+\pi)\exp[j\omega(x\cos(\theta+\pi) + y\sin(\theta+\pi))]\omega d\omega d\theta \quad (5.6-14)$$

把关系$F(\omega, \theta+\pi) = F(-\omega, \theta)$和$t = x\cos\theta + y\sin\theta$带入上式可以得到：

$$f(x, y) = \int_0^{\pi}\left[\int_{-\infty}^{\infty} F(\omega, \theta) \mid \omega \mid e^{j\omega t} d\omega\right]d\theta$$

$$= \int_0^{\pi}\left[\int_{-\infty}^{\infty} \boldsymbol{S}_\theta(\omega) \mid \omega \mid e^{j\omega t} d\omega\right]d\theta \quad (5.6-15)$$

为把上式表示为滤波反投影形式，必须把等式分成两步不同的操作。

第一步是在每一个角度θ对投影数据进行滤波，如下式：

$$Q_\theta(t) = \int_{-\infty}^{\infty} \boldsymbol{S}_\theta(\omega) \mid \omega \mid e^{j\omega t} d\omega \quad (5.6-16)$$

第二步，经滤波的投影再作反投影，以获得原始的目标函数

$$f(x, y) = \int_0^{\pi} \boldsymbol{Q}_\theta(x\cos\theta + y\sin\theta)d\theta \quad (5.6-17)$$

综上所述，只要将投影$\boldsymbol{P}_\theta(t)$在频域乘$|\omega|$，即作高通滤波，求得滤波投影$\boldsymbol{Q}_\theta(t)$，然后作反投影就可以获得$f(x, y)$。若从空间域看，把$|\omega|$的冲激响应记作$h(t)$，高通滤波为$\boldsymbol{P}_\theta(t) * h(t)$，其中$h(t) = \int_{-\infty}^{\infty} \mid \omega \mid e^{j\omega t} d\omega$，表面上看问题似乎解决了，实际上(5.6-16)式是物理上不可实现的。但若认为$\boldsymbol{S}_\theta(\omega)$是限带，在$\pm\Omega$频率外其值为0，则(5.6-16)式的积分就在$(-\Omega, \Omega)$之间可以实现了。此$\Omega$与投影

函数的采样精度有关系，若在 t 轴上以间隔 τ 对投影数据采样，τ 足够小，使之满足：$2\Omega \leqslant \frac{2\pi}{\tau}$，则没有混迭。这时滤波器的冲激响应为：

$$
\begin{aligned}
h(t) &= \int_{-W}^{W} H(\omega) e^{j\omega t} d\omega \\
&= \frac{1}{2\tau^2}\left[\frac{\sin(2\pi t/2\pi)}{2\pi t/2\tau}\right] - \frac{1}{4\tau^2}\left[\frac{\sin(\pi t/2\pi)}{\pi t/2\tau}\right]^2
\end{aligned}
\tag{5.6-18}
$$

但这样的高通滤波器有较大的过冲，可以通过加窗函数来使 f 响应在高端略低。

习　题

1. 设点扩展函数

$$
h(x, y) = \begin{cases} 1, & |x| = 0, 1, 2 \text{ 和 } |y| = 0, 1, 2 \\ 0, & \text{其他} \end{cases}
$$

若 $f(x, y)$ 定义在 $x, y = 0, 1, 2, 3$，试写出 $h_e(x, y)$ 和循环矩阵 $[\boldsymbol{h}]$。

2. 设离散信号 $f(m)$ 的后向差分为

$$g(m) = f(m) - f(m-1)$$

式中，当 $m < 0$ 或 $m \geqslant M$ 时，$f(m) = 0$。若用向量形式表示，可写成

$$\boldsymbol{g} = [\boldsymbol{h}]\boldsymbol{f}$$

试写出循环矩阵 $[\boldsymbol{h}]$。

3. 设二维图像 $f(m, n)$ 在 x 和 y 方向的后向差分为

$$g(m, n) = f(m, n) - f(m, n-1) - f(m-1, n) + f(m-1, n-1)$$

式中，当 $m, n < 0$ 或 $m \geqslant M$，或 $n \geqslant N$ 时，$f(m, n) = 0$。若用向量表示，则可写成

$$\boldsymbol{g} = [\boldsymbol{h}]\boldsymbol{f}$$

这里 $[\boldsymbol{h}]$ 是什么矩阵？写出 $[\boldsymbol{h}]$ 的分块形式，指出哪些矩阵块 $[\boldsymbol{h}_i]$ 的元素都是0元素，并写出那些有非零元素的矩阵。

4. 在连续的线性移不变系统的维纳滤波中，如果噪声与信号的功率谱比 $S_{nn}(u, v)/S_{ff}(u, v) = |H(u, v)|^2$，则估值 $\hat{f}(x, y)$ 等于什么？由 $\hat{f}(x, y)$ 直接求均方误差 e，并和式(5.2-19)比较。如果从假定的等式中得出

$n = h * f$，这个结果对不对，为什么？

5. 已知矩阵$[\boldsymbol{b}] = [b_{ij}]$，$1 \leqslant i \leqslant N$，$1 \leqslant j \leqslant M$，求函数$f$对矩阵$[\boldsymbol{b}]$的导数，定义为$\frac{\partial f}{\partial [\boldsymbol{b}]} = \left[\frac{\partial f}{\partial b_{ij}}\right]$。

(1) 已知矩阵$[\boldsymbol{a}] = [a_{ij}]$，$1 \leqslant i \leqslant M$，$1 \leqslant j \leqslant N$，求$\frac{\partial T_r[\boldsymbol{a}][\boldsymbol{b}]}{\partial [\boldsymbol{b}]}$及$\frac{\partial T_r[\boldsymbol{b}][\boldsymbol{a}]}{\partial [\boldsymbol{b}]}$；

(2) 已知矩阵$[\boldsymbol{a}] = [a_{ij}]$，$1 \leqslant i, j \leqslant M$，求$\frac{\partial T_r[\boldsymbol{b}][\boldsymbol{a}][\boldsymbol{b}]^{\mathrm{T}}}{\partial [\boldsymbol{b}]}$。

6. 设图像$f(i, j)$已经归一化，即

$$\sum_{i=0}^{M-1} \sum_{j=0}^{N-1} | f(i, j) |^2 = 1$$

将它延拓并堆叠成向量形式后每个元素为f_i，$i = 0, 1, \cdots, PQ-1$。在线性移不变系统中对应的模糊图像是使图像的熵最大，即

$$-\sum_{i=0}^{PQ-1} f_i \log f_i = E_n$$

为最大，同时满足图像的噪声能量是常数的要求，求这个问题的空间域的方程。

7. 如果$f(x, y)$是圆对称的，那么它可以由单个投影重建，为什么？

8. 如果$f(x, y)$可以分解为$f_1(x)$和$f_2(y)$的乘积，即$f(x, y) = f_1(x) f_2(y)$，试证明只需两个与坐标轴垂直的投影即可重建$f(x, y)$。

参考文献

[1] Andrews H C, Hunt B R. Digital Image Restoration [M]. Prentice-Hall, 1977.

[2] Phillips D L. A technique for the numerical solution of certain integral equations of the first kind [J]. J. Assoc. Comp. March, Vol. 9, 1962, 84~97.

[3] Hunt B R. The application of constrained least square estimation to image restoration by digital computer [J]. IEEE Trans. Computers, Vol. C22 No. 9, Sept. 1983, 805~812.

[4] D. Slepian, Restoration of photographs blurred by image motion [J]. B. S. T. J. Vol. 46, No. 10, Dec. 1967, 2357~2362.

[5] Gonzalez R C, Wintz. Digital Image Processing [M]. Addison-Wesley, 1977.

[6] 章毓晋. 图像工程(上册)—图像处理和分析[M]. 北京：清华大学出版社，1999.

[7] 周新伦，等. 数字图像处理[M]. 北京：国防工业出版社，1986.

第6章 图像数据压缩编码

数字图像的数据量十分庞大，若不进行压缩，对其进行存储或传输需要消耗大量存储容量或信道带宽资源，从而将妨碍图像信息在各领域的应用。对图像数据进行压缩编码势在必行。

利用原始图像中存在的冗余度可以实现图像压缩。近十余年来在图像压缩领域取得了巨大的进展，可以说图像压缩是数字图像处理学科中最活跃的分支。在图像压缩方面发表的论文数远超过图像分割、增强、复原等等方面的论文数。以1990年ITU-T制订的H.261视频编码国际标准为起点的序列图像编码标准的制订，标志着这项技术的成熟和大规模推广应用的开始。图像编码技术的成果对工业界和人们的日常生活产生了很大的影响，VCD，DVD，数字电视和网络多媒体……的出现都得益于图像编码技术的进展。

本章介绍图像压缩编码的基本原理和常用方法。限于篇幅，对近年来制订的各种图像编码国际标准只在§6.6中作简要的说明。

6.1 图像编码概述

6.1.1 图像压缩编码的必要性和可能性

图像的数据量非常大。为了有效地传输和存储图像，有必要压缩图像的数据量。随着现代通信技术的发展，要求传输的图像信息的种类和数据量愈来愈大。若不对此进行数据压缩，便难以推广应用。

图像数据可以进行压缩有几方面的原因。首先，原始图像数据是高度相关的，存在很大的冗余度。数据冗余造成比特数浪费，消除这些冗余可以节约码字，也就是达到了数据压缩的目的。大多数图像内相邻像素之间有较大的相关性，这称为空间冗余度。序列图像前后帧之间有较大的相关性，这称为时间冗余度。其次，若用相同码长表示不同出现概率的符号也会造成比特数的浪费，这种浪费称为符号编码冗余度。如果采用可变长编码技术，对出现概率高

的符号用短码字表示，对出现概率低的符号用长码字表示，这样就可大大消除符号编码冗余度。

允许图像编码有一定的失真也是图像可以压缩的一个重要原因。在许多应用场合，并不要求经压缩编码及解码以后所得的复原图像和原图完全相同，而允许有少量失真。只要这些失真并不被人眼所察觉，在许多情况下是完全可以接受的。这就给压缩比的提高提供了十分有利的条件。图像质量允许的损失愈多，可以实现的压缩比就愈大。这种有失真的编码常称为限失真编码。在多数应用中，人眼往往是图像信息的最终接收者(信宿)。如果能充分利用人眼的视觉特性，就可以在保证所要求的图像主观质量的前提下实现较高的压缩比。这就是利用了视觉冗余度。其实人类的视觉系统(HVS)是有缺陷的，对某些失真不敏感，难以觉察。一个图像编码方法如果能充分利用这些特性，就可取得较好的效果，即在复原图像主观质量较好的前提下得到较高的压缩比。我们举几个例子来说明：

(1) 人眼对颜色的空间分辨率低于对亮度信号的空间分辨率。因而在对彩色图像编码时，可以先通过色空间转换把 RGB 空间转换为 Y，C_b，C_r 空间，并对 C_b，C_r 分量作亚抽样，降低色差信号的空间分辨率，然后作压缩编码。以此来提高压缩比，而仍能获得较高主观质量的复原图像。

(2) 人眼对画面静止部分的空间分辨率高于活动部分。所以，可以对静止部分减少时间分辨率，而对快运动部分减少空间分辨率，达到数据压缩的目的。

(3) 人眼不能觉察亮度的细小变化。即存在视觉阈值，而且此阈值随着图像内容的变化而变化。在平坦区，视觉阈值低，对量化失真较敏感。在边缘和纹理区，对量化失真不敏感，这就是视觉掩盖效应。这种特性被广泛用来提高压缩比。

(4) 人眼对屏幕中心区的失真敏感，对屏幕四周的失真不敏感。因此对四周的粗糙量化也可以节约码字。

此外，还可以利用先验知识实现图像编码。在某些特定的应用场合，编码对象的某些特性可预先知道。例如，在可视电话中，编码对象为人的头肩像。这时，可以利用对编码对象的先验知识为编码对象建立模型。通过提取模型参数，对参数进行编码而不对图像直接进行编码，可以达到非常高的压缩比。这是模型编码的基本思想，这里利用了知识冗余度。

图像编码技术就是把种种压缩的可能性变为现实的技术。

6.1.2 图像压缩编码的一般框图

图像压缩编码的一般过程可以概括成图 6.1 所示的三个步骤。原始图像先经

过映射变换，然后再经量化和熵编码器成为码流输出。

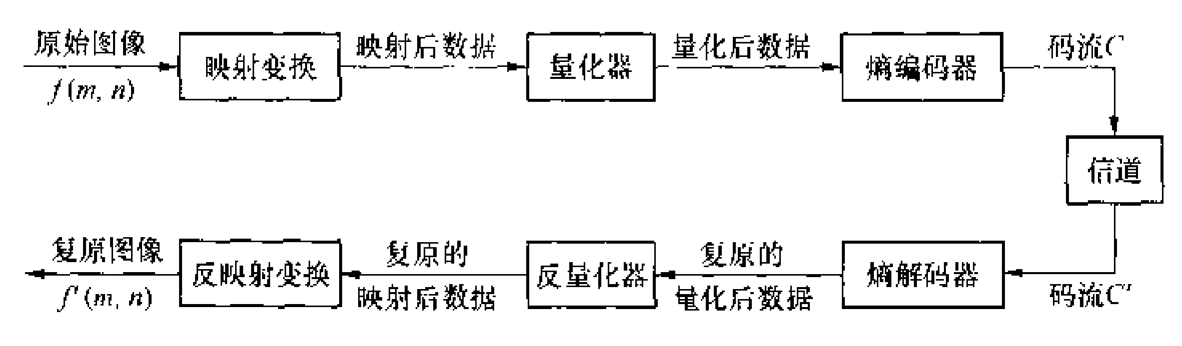


图6.1 图像压缩编码的一般框图

现对编码框图作下列说明：

1. 映射变换

其目的是通过映射改变图像数据的特性，使之更有利于压缩编码。

例如以左邻像素灰度值作为当前像素灰度值的预测值，以预测差作为映射后数据，并对其进行量化编码。由于求差值的过程减少了相邻像素间的相关性，从而减少了冗余度，因此可以实现压缩。

如果采用更好的预测方法，如由空间相邻像素点预测当前像素值（帧内DPCM），或用前一帧的像素值预测当前像素值（帧间DPCM），可望得到更准的预测值，从而进一步提高压缩比。在实际应用中，映射变换的方法种类繁多，还可能更复杂。如：在变换编码中，先将图像分成若干个$n \times n$大小的子块，并对各子块进行某种正交变换，而量化和编码是对变换后所得系数进行的。事实上，映射变换是图像编码的一个核心部分，它决定了量化和编码的对象类型。所以可以据此对编码方法进行分类。例如，若映射变换中进行的是帧内预测，则称其为帧内DPCM编码方法，若进行的映射变换是帧间预测，则称其为帧间DPCM编码。二者都属于波形编码，因为是直接在空间域中对图像灰度值或其简单运算后的结果进行量化编码的。而变换编码则是对经映射后得到的变换域中的数据进行量化编码的。在模型编码中，量化编码的对象是模型参数。这里的映射变换实际上是一个复杂的建模和对模型参数进行估计的过程。它把原始图像映射成模型参数，从而实现了很高的压缩比。从这些例子可以体会到映射变换在图像编码中的核心作用。除模型编码外，映射变换一般没有直接压缩数据。一个好的映射变换与合适的量化器和编码器相配合，应能较好地消除图像信源的各种冗余度。

2. 量化器

在限失真编码中要对映射后的数据进行量化。若量化是对映射后的数据逐个地进行的，则称标量量化。若量化是成组地进行的，则称矢量量化。量化可以直接减少原始数据量，从这种意义说，量化本身即可起到压缩数据的作用。但量化总会

造成原始数据信息的丢失，形成失真，这称之为量化失真或量化噪声。为使量化失真小，应量化得精细，这样压缩比就高不了，这是一对矛盾。应选用恰当的量化级数和量化曲线形状来缓解这对矛盾。量化器的引入是图像编码产生失真的根源。在要求复原图像与原图完全一致的无失真编码器中必须不用量化器。但这样一来，压缩比难以提高。在多数应用中，允许存在少量失真，只要把失真的程度和性质控制在允许的范围内。也就是把复原图像的主观质量控制在允许的程度内，就可以在满足应用要求的前提下提高压缩比。值得注意的是，对于同样的量化失真，由不同的映射变换与反变换会引起不同性质的复原图像的失真，人眼对某些性质的失真敏感而对另一些性质的失真不敏感。

3. 熵编码器

这一步是用来消除符号编码冗余度的，它一般不产生失真。理想的情况是使编出码流的平均码长等于量化后数据的信息熵。常用的编码方法有许多种，例如分组码中的行程码(RLC)、变长码(VLC)以及不分组码中的算术码(Arithmatic Coding)等等。

上述三个步骤之间是相互联系相互制约的。对有些编码方法，如预测编码或变换编码，映射变换后数据量并没有减少，甚至因动态范围的加大而使数据量略有增加。但它为后两步作了准备，使它们能有效发挥作用。而在模型编码中，经映射变换后得到的模型参数，其数据量已大大小于原始图像，即第一步已实现了很大的压缩。后面的量化编码则是作进一步的压缩。其情况和波形编码、变换编码有很大的不同。我们举一个例子来说明各部分之间的相互影响：如果第二步使用了不均匀量化器，它可使每层量化值的出现概率接近相等，那么后面再用变长码，这第三步的压缩效果就会大打折扣。或者反过来说，若后面用了变长码，前面的量化器也可以不妨采用线性量化器，总的效果仍然很好，量化器实现起来也要方便得多。

此外，如果码流在信道中传输时产生误码，就会造成复原图像质量的急剧下降。有时一个比特的误码会造成一大片复原图像的错误，这种现象称误码扩散现象。例如，在上述帧内预测编码例子中，由于编码端是对当前像素与左邻像素的差值进行量化编码，则在解码端，若因误码造成了某个像素复原值有误，则从该像素起，以后复原的所有像素都有误。而且，值得注意的是，误码可能导致码流结构的变化，造成不能正确解码，从而加剧误码扩散现象。按信息论的观点，压缩得越多、冗余度越小的码流越难抵抗误码的影响。在实用的图像编码算法中必须对误码扩散提出有效制约对策。

6.1.3 图像压缩编码的分类

图像编码应用广泛，有许多编码方法。对它们进行分类有助于对问题的理解

和解决，基于不同的角度，有许多种分类方法。

1. 按复原图像是否与原图一致分类

可将编码方法分为两大类：无失真编码和限失真编码。无失真编码又称信息保持编码(Lossless Coding)或可逆编码。限失真编码又称非信息保持编码(Lossy Coding)或不可逆编码。此时，编码会造成失真，不过这些失真可以被控制在一定的限度内，不致影响使用效果。在无失真编码中不可使用量化器，因为量化总会带来不可恢复的失真。无失真编码的压缩比低，可达到的最高压缩比取决于信源熵。在限失真编码中允许的失真愈大，可达到的压缩比就愈高，它与量化的粗细有很大关系。在限失真编码中把失真控制在视觉阈值以下或控制在可容忍的程度是有效地提高压缩比的关键。

2. 按待压缩图像的不同属性分类

例如：以图像的光学特征分，有单色、彩色、多光谱图像的压缩编码；以灰度等级分，有二值图、多值图与灰度图像编码；若以动静来分，有静止图像编码与序列图像编码之分，后者也称视频编码或活动图像编码。对不同要求、不同性质的图像用适当的编码方法和编码参数进行压缩编码是达到预期目标的关键。例如，在静止图像编码中，只能进行帧内编码，而且人眼对静止图像的失真较活动图像更为敏感，压缩比一般没有活动图像编码来得高。但活动图像编码常采用运动估计(ME)和运动补偿(MC)技术，并且对实时性的要求很高，故要求很高的运算速度，这是它特有的难点。

3. 按所用方法的原理分类

可将图像编码方法分为基于图像统计特性、基于人眼视觉特性和基于特征提取等方法。在实际编码中，常常要同时用到图像的统计特性和人眼视觉特性才能进行有效的编码，难以把它们截然分开。而有些编码方法的类别还可进一步细分，如基于统计特性的编码方法还可分为帧内预测、帧间预测、变换编码等。许多新的图像编码国际标准使用了混合编码方法，即同时用到了帧内、帧间预测和变换编码。

也有文献按映射变换实现方法来进行分类，即按照究竟对什么性质的数据进行量化编码来分类。按此方法可分为四大类：

(1) 波形编码。帧内预测、帧间预测编码都属于波形编码。

(2) 变换编码。此时，映射变换是某种正交变换，如分块 DCT 变换等等。

(3) 模型编码。此时的映射变换是个模型的建立与模型参数估计的过程。它的实现相当难，因此，虽然它有很高的压缩比，但在对自然景物图像编码时，目前还很少采用模型编码。

(1) 拟合编码。此时的映射变换是一个曲面、曲线拟合的过程。属于这类的编码方法有方块截尾编码(Block Truncating Coding, BTC)和分形编码等。

6.2 图像压缩编码的理论基础

本节分别讨论无失真编码及限失真编码的基本理论,讨论中要引用信息论中的一些结论。

6.2.1 无失真编码理论基础

无失真编码又称信息保持编码或熵编码。它要求经过编解码后复原的图像和原图完全一样,编码过程不丢失任何信息。如果对已量化的信号进行编码,必须注意到量化所产生的失真是不可挽回的。所以,这里的无失真是对已量化的信号而言的。

下面讨论熵与相关性及冗余度的关系,这里首先不加证明地介绍信息论中的一个定理——香农的无干扰编码定理。在无干扰的条件下,存在一种无失真的编码方法,使编码的平均长度 $\bar{L}(x)$ 与信源的熵 $H(x)$ 任意地接近。这就是说,无失真编码的平均码长存在一个下限,这个下限就是原始图像的熵。

在这个定理的基础上可以定义某种编码方法的效率为:

$$\eta \triangleq H(x)/\bar{L}(x)$$

冗余度为:

$$r \triangleq 1-\eta = [\bar{L}(x) - H(x)]/\bar{L}(x)$$

为了传输或存储一个像素平均需要的比特数称为比特率。若原始图像的平均比特率为 n,压缩编码后降低为 n_d,则压缩比定义为:

$$C \triangleq n/n_d$$

由上述定理,n_d 不可能比 $H(x)$ 小,故无失真编码可以达到的最大压缩比为

$$C_{max} = n/H(x)$$

现在先看独立信源的熵。独立信源又叫无记忆信源,其特点是某个位置上出现的符号与其他位置上出现什么符号无关。设信源的符号表为 $\{x_1, x_2, \cdots, x_q\}$,各符号出现的概率为 $\{P(x_1), P(x_2), \cdots, P(x_q)\}$,则此信源的熵为

$$H(x) = -\sum_{i=1}^{q} P(x_i)\log_2 P(x_i) \tag{6.2-1}$$

设 $q=2^L$。若独立信源符号表中各符号出现的概率相等,均为 $1/2^L$,则其熵为最大,等于 L 比特。这时,用普通的等长自然二进制码来编码就已经达到编码效率为1,对它无法作数据压缩。在不等概率分布时,熵 $H(x)$ 小于 L,若采用和概率分布相适应的不等长编码可以使平均码长 $\bar{L}<L$,实现数据压缩。

例如,有独立信源 X,其符号表为 $\{x_1, x_2, x_3, x_4\}$,其出现概率分别为 $P(x_i)$,$i=1,2,3,4$。用 $C(x_i)$ 表示对它们的编码,用 $L(x_i)$ 表示其码长。表6.1列出各符号等概率出现及不等概率出现两种情况下用不同编码方法所得到的编码效率。表中第3行表明,在各符号以等概率出现时用等长自然二进码 C_1 已经达到效率为1。第4行表明当 x_i 不等概率出现时,若仍用等长码 C_1 会降低编码效率。若改用表中第5行所示的不等长码 C_2,以长度短的码代表出现概率大的符号,则可减少平均码长,提高效率。由表6.1中第7行可以看出,这样做了以后效率又提高为1。

表 6.1 不同情况下的编码效率

x_i	x_1	x_2	x_3	x_4	$\bar{L}(x)$	$H(x)$	η	r	注
$C_1(x_i)$	00	01	10	11					
$L_1(x_i)$	2	2	2	2	2				
$P_1(x_i)$	1/4	1/4	1/4	1/4		2	1	0	等概信源,等长码
$P_2(x_i)$	1/2	1/4	1/8	1/8		1.75	0.875	0.125	不等概信源,等长码
$C_2(x_i)$	0	10	110	111					
$L_2(x_i)$	1	2	3	3	1.75				
$P_2(x_i)$	1/2	1/4	1/8	1/8		1.75	1	0	概率小,用长码

下面再来分析有限马尔可夫信源的熵。先看一维情况,m 阶马尔可夫信源的符号 x_i 的出现概率仅与前面 m 个符号有关,其统计特性要用条件概率来描述。设信源符号表为 $X=\{x_1, x_2, \cdots, x_q\}$,条件概率为 $P(x_n \mid x_{n-1}, x_{n-2}, \cdots, x_{n-m})$ 这里下标 n 表示当前时刻,下标 $n-i$ 表示前第 i 个时刻,而 x_n 表示当前时刻符号的状态。例如 $n=3$,$X=\{0,1\}$ 时,码110可以用 $x_{n-2}x_{n-1}x_n$ 来表示,现在 $x_{n-2}=1$,$x_{n-1}=1$,$x_n=0$。在已知 $x_{n-1}, \cdots, x_{n-m}$ 状态以后再接收到符号 x_n 所得到的信息量为 $I(x_n \mid x_{n-1}, \cdots, x_{n-m})=-\log_2 P(x_n \mid x_{n-1}, \cdots, x_{n-m})$ 平均信息量为

$$I(x_n \mid x_{n-1}, \cdots, x_{n-m})=-\sum_{x_n=x_1}^{x_q} P(x_n \mid x_{n-1}, \cdots, x_{n-m})\log_2 P(x_n \mid x_{n-1}, \cdots, x_{n-m})$$

再考虑到 $x_{n-1}, \cdots, x_{n-m}$ 序列发生的概率,可以得到条件熵

$$H(x_n \mid x_{n-1}, \cdots, x_{n-m}) = -\sum_{x_{n-1}=x_1}^{x_q} \cdots \sum_{x_{n-m}=x_1}^{x_q} \left[\sum_{x_n=x_1}^{x_q} P(x_{n-1}, \cdots, x_{n-m}) \cdot P(x_n \mid x_{n-1}, \cdots, x_{n-m}) \log_2 P(x_n \mid x_{n-1}, \cdots, x_{n-m})\right]$$
$$= -\sum_{x_n} \cdots \sum_{x_{n-m}} P(x_n, x_{n-1}, \cdots, x_{n-m}) \log_2 P(x_n \mid x_{n-1}, \cdots, x_{n-m}) \tag{6.2-2}$$

从联合概率的角度出发,可以定义信源的高阶熵。只看两符号同时出现的概率可以定义二阶熵。

$$H_2 = H(x_n, x_{n-1}) = -\sum_{x_n=x_1}^{x_q} \sum_{x_{n-1}=x_1}^{x_q} P(x_n, x_{n-1}) \log_2 P(x_n, x_{n-1}) \tag{6.2-3}$$

由联合概率 $P(x_n, \cdots, x_{n-m})$ 可以得到 $m+1$ 阶熵

$$\begin{aligned} H_{m+1} &= H(x_n, x_{n-1}, \cdots, x_{n-m}) \\ &= -\sum_{x_n} \cdots \sum_{x_{n-m}} P(x_n, \cdots, x_{n-m}) \log_2 P(x_n, \cdots, x_{n-m}) \end{aligned} \tag{6.2-4}$$

它是长度为 $m+1$ 的序列的联合熵,是对 $m+1$ 个符号同时编码所需比特数的下限。条件熵和联合熵之间有下述关系

$$H(x_n \mid x_{n-1}, \cdots, x_{n-m}) = H_{m+1} - H_m \tag{6.2-5}$$

现在来证明上式。由条件概率的性质

$$P(x_n, x_{n-1}, \cdots, x_{n-m}) = P(x_n \mid x_{n-1}, \cdots, x_{n-m}) P(x_{n-1}, \cdots, x_{n-m})$$

代入(6.2-2)式得

$$H(x_n \mid x_{n-1}, \cdots, x_{n-m})$$
$$= -\sum_{x_n} \cdots \sum_{x_{n-m}} P(x_n, x_{n-1}, \cdots, x_{n-m}) \log_2 \frac{P(x_n, x_{n-1}, \cdots, x_{n-m})}{P(x_{n-1}, \cdots, x_{n-m})}$$
$$= -\sum_{x_n} \cdots \sum_{x_{n-m}} P(x_n, x_{n-1}, \cdots, x_{n-m}) \log_2 P(x_n, x_{n-1}, \cdots, x_{n-m})$$
$$-\left\{-\sum_{x_{n-1}} \cdots \sum_{x_{n-m}} \left[\sum_{x_n} P(x_n, x_{n-1}, \cdots, x_{n-m})\right] \cdot \log_2 P(x_{n-1}, \cdots, x_{n-m})\right\}$$

由于

$$\sum_{x_n} P(x_n, x_{n-1}, \cdots, x_{n-m}) = P(x_{n-1}, \cdots, x_{n-m}),$$

再由(6.2-4)式可得 $H(x_n \mid x_{n-1}, \cdots, x_{n-m}) = H_{m+1} - H_m$，证毕。

下面我们对上述公式的含义作进一步的说明。为便于理解，先看二阶情况，然后推广至 m 阶。当 x_n 与 x_{n-1} 完全独立时，

$$\begin{aligned} H_2 &= -\sum_{x_n}\sum_{x_{n-1}} P(x_n)P(x_{n-1})[\log_2 P(x_n) + \log_2 P(x_{n-1})] \\ &= -\sum_{x_n} P(x_n)\log_2 P(x_n) - \sum_{x_{n-1}} P(x_{n-1})\log_2 P(x_{n-1}) \\ &= H_1 + H_1 = 2H_1 \end{aligned}$$

这种情况可以推广到当 m 个符号相互独立时 $H_m = mH_1$。当 x_n 与 x_{n-1} 完全相关时

$$P(x_n, x_{n-1}) = \begin{cases} P(x_{n-1}), & x_n = x_{n-1} \text{ 时} \\ 0, & x_n \neq x_{n-1} \text{ 时} \end{cases}$$

则

$$\begin{aligned} H_2 &= -\sum_{x_n}\sum_{x_{n-1}} P(x_n, x_{n-1})\log_2 P(x_n, x_{n-1}) \\ &= -\sum_{x_{n-1}=x_1}^{x_n} P(x_{n-1})\log_2 P(x_{n-1}) = H_1 \end{aligned}$$

上式同样可推广为当 m 个符号完全相关时，$H_m = H_1$。从道理上很容易想象上述结论，当 m 个符号相互独立时，发 m 个符号所传递的信息为发单个符号时的 m 倍。若 m 个符号全相关，由收到的第一个符号已经完全可以猜到后面 $m-1$ 个符号是什么，那么收到后面的符号不会再增加信息量。因此发这种重复 m 次的信号和发单个符号所传递的信息是一样的，故有 $H_m = H_1$。一般情况介于这两种极端情况之间，有

$$mH_1 \geqslant H_m \geqslant H_1 \tag{6.2-6}$$

当相邻符号间相关性加大时，H_m 趋近于 H_1；反之趋近于 mH_1。对图像来说，其像素之间的相关性究竟如何？我们列出一组典型数据：$H_1 = 7$ bit，$H_2 = 11$ bit，$H_3 = 14$ bit，$H_4 = 16$ bit，$H_5 = 17$ bit。可以看出图像高阶熵的增加量愈来愈少。这就是说，相邻像素之间有较大的相关性。如果已知图像的一些像素后，要确定未知像素所需的码长（和熵对应）是比较小的。图像是二维信源，它与随机场相对应。要仔细分析比上述对随机序列的分析更麻烦，但上述结论指出的趋势是对的。利用像素间相关性减少码长的具体做法将在下面几节

中详细讨论。

6.2.2 限失真编码理论基础

图像的失真怎么度量，至今还没有一个很好的度量标准。在人眼判读的情况下，人眼是图像质量的最终评判者。但是对人眼视觉机理至今尚未完全掌握，所以难以得到一个和主观评价十分相符的客观评价标准。目前用得最多的仍是均方误差。设原图为 $f(m, n)$，失真图像为 $g(m, n)$，则失真为

$$D = \frac{1}{MN}\sum_{m=0}^{M-1}\sum_{n=0}^{N-1}[f(m, n) - g(m, n)]^2 \qquad (6.2-7)$$

这个失真度量标准并不好，之所以用得多，主要是为了方便。具有同样均方误差的两幅失真图像，因失真部位不同，失真性质不同，其主观评价可以有很大的差别。因此均方误差 D 绝不能和图像质量等同看待。事实上，寻求一种和图像的主观评价相符的客观失真度量标准，仍然是一项艰巨而重要的任务。实用中常用的 PSNR 值与这里的 D 之间的关系可参见式(2.6-1)。

上小节讨论的无失真编码其压缩比一般是不大的。在许多应用中为了进一步压缩数据，允许存在少量失真。现在的问题是如果允许失真不得超过某个值 D，进行压缩编码的比特率有没有一个下限？若有，这个下限是什么？信息论中的率失真理论回答了这个问题。根据率失真理论，当信息在有噪声信道中传输时，有一个函数 $R(D)$ 存在，只要信道容量不小于 $R(D)$，就可以传输信息，使失真小于或等于 D。这个函数称为率失真函数。可以把含量化器的信源编码器看成一个有噪信道(含量化噪声)，并根据信息论中有关信息在有噪信道中传输的一些结论来推导出平均传输一个像素至少需要的比特数。信息论表明，传输信道容量应大于等于某个下限时，才能以小于 D 的失真传输信息。那么类似地，可推出在编码时平均每个像素至少用多少个比特编码才能使编码造成的失真小于预定的 D。

1. 率失真函数 $R(D)$

这里 D 是指允许的失真。D 可以用均方误差，也可以用其他标准度量。$R(D)$ 便是在允许失真 D 下信源编码所需的最小平均信息量，在这里可认为是比特率。

图 6.2 是一个离散信源率失真函数的示意图。可以看到，当 $D=0$，即无失真编码时，需要的比特数为 $R(0)$。随着允许的失真 D 逐渐增大，所需的比特数也下降，并可证明此关系曲线是下凸的。一般 D_{max} 等于信号方差 σ^2。当 $D \geq D_{max}$，即允许的失真超过了输入信号的方差后，此时传输数据信息已失去意义，不需要对信

道有什么要求了，$R(D)=0$。

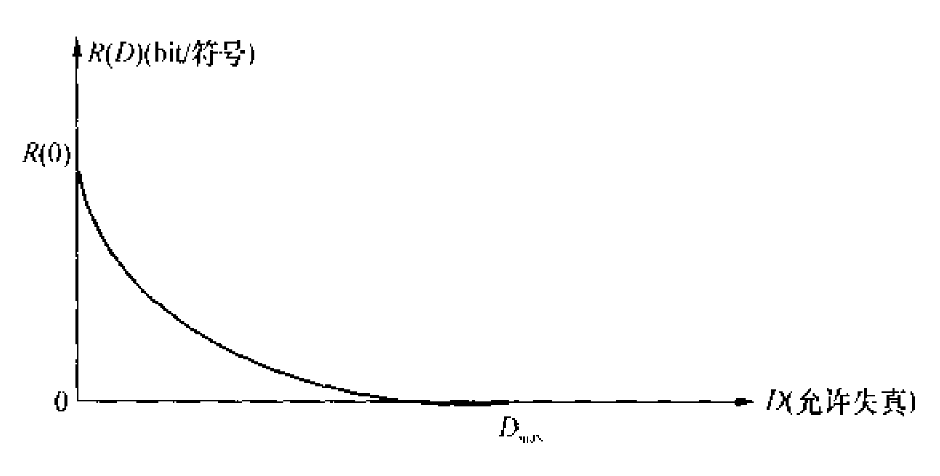


图 6.2　离散信源的 D 与 $R(D)$ 的关系曲线

2. *信源编码定理*

对一个率失真函数为 $R(D)$ 的信源，若平均失真为 D，则必存在一种编码方法，使比特率 $R>R(D)+\delta$，而平均失真 $D\leqslant D+\varepsilon$，其中 ε，δ 为任意小的正数。

率失真函数与失真度量标准及信源统计特性有十分密切的关系，信源不确定性大，方差大，则所需比特数 $R(D)$ 较大，信源平稳、变化小、方差小，则 $R(D)$ 也小。可以证明，对于以均方误差作为失真度量的正态分布信源，率失真函数为

$$R(D)=\begin{cases}\dfrac{1}{2}\log_2\dfrac{\sigma^2}{D}, & 0\leqslant D\leqslant\sigma^2\\ 0, & D>\sigma^2\end{cases}\tag{6.2-8}$$

由上式，在对图像数据做映射变换时，应使映射后数据的 σ^2 尽量小。

本节仅从信息论的观点进行讨论，在实际应用中，还可利用其他特性来提高信源编码的性能。例如，利用人眼视觉特性和人的先验知识等。

6.3　熵　编　码

本节介绍各种以消除符号编码冗余度为目标的编码方法。主要介绍变长码(Variable Length Coding，VLC)和算术码(Arithmetic Coding)。其中变长码主要介绍霍夫曼码，同时对可逆变长码(RVLC)和通用变长码(UVLC)作简单介绍。本节介绍的各种符号编码方法在实现数据压缩的同时并不引起失真，但会降低其抗误码能力。由 6.2.1 节的讨论可知，它们能实现的比特率下限就是信源的信息熵，常用于图像的无失真编码中。在限失真编码中则用作图 6.1 框图中的熵编码，以

消除量化后数据的符号编码冗余度，在不增加失真的前提下对数据作进一步地压缩。

6.3.1 霍夫曼(Huffman)码

由上节所举的例子可以看到，当符号表中各符号出现概率不相等时，若采用不等长码字，对概率大的符号赋予短的码字，对概率小的符号赋予长的码字，从而可以减少平均码长，提高编码效率。霍夫曼码就是基于这种思想的码。在各种变长码中，霍夫曼码是冗余度最小的编码，即其平均码长最接近于信源的熵。虽然它是不等长码，但在收端可以唯一地解码，且不造成失真。现在来看霍夫曼码的编码过程，这是一个由最后一位码向前逐位确定的过程，要点如下：

(1) 把符号按出现概率排序，概率大的在前。给最后的两个符号赋予相同的码长（长度未知），这两个符号的码只有最后一位不同，其他各位都一样。这一步只确定了概率最小的两个符号的最末一位码，前面几位码字的内容要在后续步骤中确定。

(2) 把最后两个符号的概率加起来合成一个新的概率，再按概率大小重新排序。在合并过程中总是只对最后两行赋以相同的码长，与这两行对应的码只有最后一位不同，前面各位都相同，即每一步都只确定最末两行的最后一位码。

(3) 重复步骤(2)，直到最后只剩下两个概率为止。

表6.2列出表6.1中第5行$C_2(x_i)$的编码过程。对照表6.2不难了解上述三个步骤的具体做法与含义。

表6.2 表6.1中$C_2(x_i)$的编码过程

原始信号及其概率			第一步			第二步			最终结果	
x_i	$P(x_i)$	码	x_i	$P(x_i)$	码	x_i	$P(x_i)$	码	x_i	码
x_1	$\frac{1}{2}$		x_1	$\frac{1}{2}$		x_1	$\frac{1}{2}$	0	x_1	0
x_2	$\frac{1}{4}$		x_2	$\frac{1}{4}$	×0	x_{234}	$\frac{1}{2}$	1	x_2	10
x_3	$\frac{1}{8}$	××0	x_{34}	$\frac{1}{4}$	×1				x_3	110
x_4	$\frac{1}{8}$	××1							x_4	111

表6.3为霍夫曼编码过程的另一个例子。对于这个例子，可以算得霍夫曼码的平均码长为：

$$L = \sum_{i=1}^{8} P(x_i)L(x_i) = 2.64$$

熵
$$H(x) = -\sum_{i=1}^{8} P(x_i)\log_2 P(x_i) = 2.56$$

效率
$$\eta = H(x)/\overline{L}(x) = 0.97$$

在接收端,可将霍夫曼码用树型解码法唯一地解码。先根据编码的码表形成解码树,解码时从高位到低位依次输入,每输入一位即可确定分支情况。与表 6.3 对应的霍夫曼码的解码树及解码过程如图 6.3 所示。

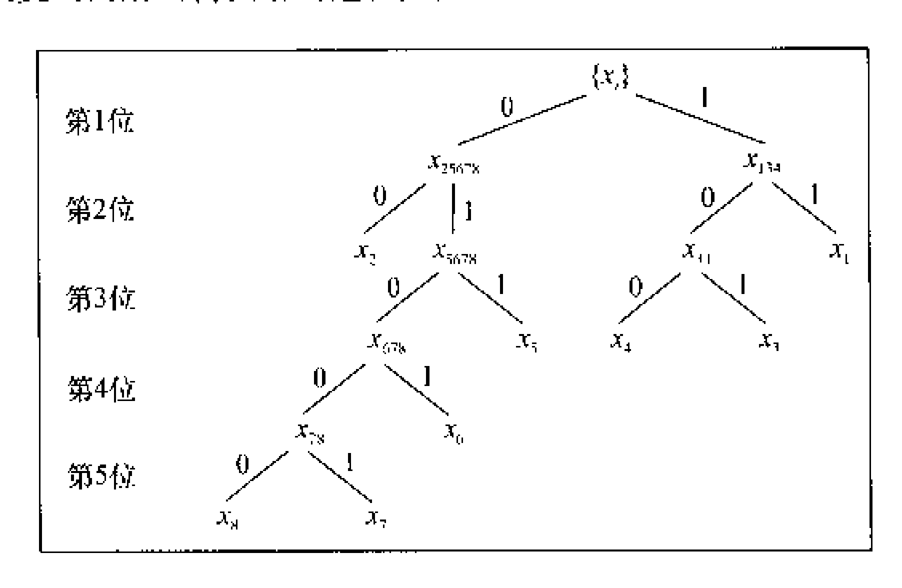


图 6.3　霍夫曼码树形解码一例

例如,收到的码为 1110101001,第一位码是 1,它在图 6.3 中取右分支;下面一位又为 1,又取右分支,到达树的叶节点 x_1。可见收到码的第 1, 2 位表示 x_1。而第三位必然是新码字的起始位,容易看出第 3, 4, 5 位的 101 表示 x_3,第六位又是下一个码字的起始位。与上面类似,由图 6.3 的树可以知道第六至十位表示 x_7。于是用树形解码方法由 1110101001 解得 $x_1x_3x_7$。从这个解码过程看出,虽然霍夫曼码是不等长码,但在解码过程中能自己确定码字的起止位,解码结果是唯一的,不会因起始位难以确定而造成任何误解或混淆。也就是说霍夫曼码具有唯一可译码性。仔细看霍夫曼码的码表,码表中任何一个短码码字都不是码表中长度更长的码字的前缀。具有这种特性的码称为前缀码。所有前缀码都有唯一可译码性(UD 特性)。从霍夫曼码编码树的结构来看,码表中所有码字均是编码树的叶节点。这就保证了霍夫曼码是一种前缀码,从而保证了它一定具有唯一可译码性。只要没有误码,解码端在变长码的码字分断中不会搞错。但若存在误码,则解码器的分断就会出现混乱。一个比特的错误可能会引起一串符号的解码错误。下小节介绍的可逆变长码可以减轻由误码造成的信息丢失。

表 6.3 霍夫曼码编码过程举例

原始信号及概率			第一步			第二步			第三步			第四步			第五步			第六步			编码结果	
x_i	$P(x_i)$	码	x_i	$P(x_i)$	码	x_i	$P(x_i)$	码	x_i	$P(x_i)$	码	x_i	$P(x_i)$	码	x_i	$P(x_i)$	码	x_i	$P(x_i)$	码	x_i	码
x_1	0.35		x_1	0.35		x_1	0.35		x_1	0.35		x_1	0.35		x_{25678}	0.37		x_{134}	0.63	1	x_1	11
x_2	0.15		x_2	0.15		x_2	0.15		x_{5678}	0.22		x_{34}	0.28		x_1	0.35	×1	x_{25678}	0.37	0	x_2	00
x_3	0.15		x_3	0.15		x_3	0.15		x_2	0.15		x_{5678}	0.22	×1	x_{34}	0.28	×0				x_3	101
x_4	0.13		x_4	0.13		x_4	0.13		x_3	0.15	××1	x_2	0.15	×0							x_4	100
x_5	0.12		x_5	0.12		x_5	0.12	××1	x_4	0.13	××0										x_5	011
x_6	0.06		x_6	0.06	×××1	x_{678}	0.10	××0													x_6	0101
x_7	0.03	×…×1	x_{78}	0.04	×××0																x_7	01001
x_8	0.01	×…×0																			x_8	01000

下面介绍霍夫曼码的并行解码。用树形解码器需进行一位一位地读入和判断，对不同长度的码字，所需解码时间不同，不适宜于用硬件实时解码。而并行解码法则可以用统一的操作步骤一次解出一个码字，易于硬件实现。

并行解码步骤如下：

(1) 一次读入 L_{max} 个比特。这里 L_{max} 为最长码字的长度。它可能为一个最长的码字，也可能为1个短码字，后面跟着一些比特，总之，它总包含一个完整的码字。

(2) 查并行解码器码表，解出第一个码字所表示的符号及其码长 L。表6.4是一个与表6.2所示的Huffman码表相对应的并行解码器码表。

(3) 将比特流左移 L 位，即丢去已解码的 L 位后重新再取较高位的 L_{max} 比特，重复上面的解码过程。

当比特流为011010111时，并行解码法解码过程如下：

本例中 L_{max} 为3 bit。首先读入高位的3 bit 即011，从表6.4中可查得码字为 x_1，查码长表可知 x_1 码长 $L=1$，于是左移1 bit，继续考察较高位的110，解出 x_3 及长度 $L=3$，再左移3 bit，考察101，解出 x_2，最后解出 x_4。于是得出收到的码字串对应符号为 $x_1x_3x_2x_4$。

总之，并行解码一次检查 L_{max} 个比特，并解出一个码字，操作统一（查表后左移若干位），便于硬件实现。

表6.4 并行解码器码表举例，$L_{max}=3$

比特串	第一个码字	第一个码字长度	比特串	第一个码字	第一个码字长度
0 0 0	x_1	1	1 0 0	x_2	2
0 0 1	x_1	1	1 0 1	x_2	2
0 1 0	x_1	1	1 1 0	x_3	3
0 1 1	x_1	1	1 1 1	x_4	3

下面对前缀码编码效率作进一步的讨论。从上面霍夫曼码编码的两个例子可以体会：若信源 $X=\{x_1, x_2, \cdots, x_i, \cdots, x_k\}$ 的各符号发生的概率 $P(x_i)$ 都为 2^{-n_i}，这里 n_i 为正整数。则霍夫曼码的编码效率可以达到1。即在这种特殊的概率分布情况下，符号 x_i 所携带的信息量正好为整数个比特，则可以通过正确地设计码表，使代表符号 x_i 的码字 $C(x_i)$ 的长度 $L(x_i)$ 正好等于 n_i。以此来达到平均码长 $\bar{L}$ 等于熵 $H(X)$ 的平均码长的下限。若各 $P(x_i)$ 中有些符号的 $-\log_2 P(x_i)$ 不为整数，则用整数个比特的码字 $C(x_i)$ 来表示它而又要达到100%效率是不可能的。但是可以证明它有上确界 $H(X)+1$。事实上，不管 $-\log_2 P(x_i)$ 是否为整数，用刚好比它大的整数个比特的码字来表示 x_i 总是可以做到的。即 $L(x_i) < -\log_2 P(x_i)+1$。

不等式两边乘 $P(x_i)$ 并对所有符号求和就有

$$\overline{L} < H(X) + 1$$

而 $\overline{L}$ 总应 $\geqslant H(X)$。从而有下述定理：

给定离散无记忆信源 X，其熵为 $H(X)$，则存在下列可能性：

信源符号 $x_1, \cdots, x_L$ 可以被赋以二进制前缀码，它的平均码长在下式给出的上下确界之间：

$$H(X) \leqslant \overline{L} < H(X) + 1 \tag{6.3-1}$$

6.3.2 可逆变长码和通用变长码

1. 可逆变长码(RVLC)

普通的变长码在有传输误码时会因为不正确的码字分断而不能被正确解码。一种提高抗干扰能力的方法是在码流中不时周期性地插入重新同步码字。如图 6.4 所示，一旦发现错误，解码器就转入搜索下一个重新同步码字状态。在找到重新同步码字后，恢复正常解码状态。从而把误码影响限制在下一个重新同步码字之前并使系统恢复正常工作。但图中阴影区这一段码流所携带的信息将因为解码端无法正确分断码字而丢失。若采用的变长码是可逆变长码，则可以从下一个重新同步点起，沿反方向进行解码，以尽可能地恢复丢失的信息。在 RVLC 中，"可逆"的含义即是码流也可以由尾部向头部反方向地进行解码。

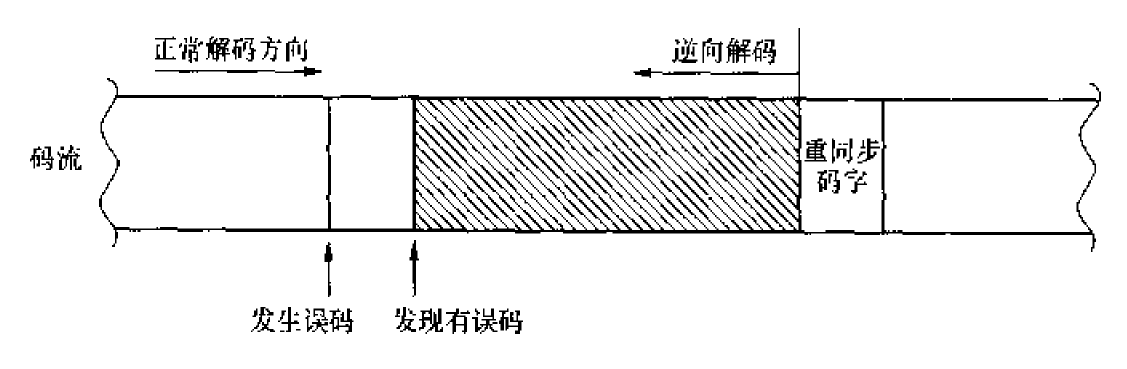


图 6.4 发现误码后的重新同步和逆向解码

霍夫曼码一般不具有可逆向正确解码的性质。以表 6.2 简例中的码表为例。从正向看(从左至右看)，短码不是长码的前缀，从而保证了正向解码时的唯一可译码性。从反向看(从右至左看)，x_1 是 x_2 及 x_3 的前缀，即它不具备反向解码时的唯一可译码性。到目前为止，还没有一个构建高效率的可逆变长码的系统方法。表 6.5 是 RVLC 的两个例子，它们都有两个方向的唯一可译码性。本例中霍夫曼码的平均码长为 2.19，对称 RVLC 的平均码长为 2.46，而不对称 RVLC 的平均码长

为2.37。即牺牲了编码效率换得了抗干扰性的提高。

表6.5　可逆变长码举例

符号 x_i	概率 $P(x_i)$	霍夫曼码表	对称RVLC	不对称RVLC
x_1	0.33	0 0	0 0	0 0
x_2	0.3	0 1	1 1	0 1
x_3	0.18	1 1	0 1 0	1 0
x_4	0.1	1 0 0	1 0 1	1 1 1
x_5	0.09	1 0 1	0 1 1 0	1 1 0 1 1

2. 通用变长码(UVLC)

通用变长码由“类别码”后接“附加码”构成。表6.6示出通用变长码的结构,它由 M 比特的0,后接一比特的1作为类别码,再接 M 比特的自然二进制码作为附加码。表6.7是依此结构构造的UVLC码表。若信源各符号的概率分布接近于Colomb分布,则用UVLC编码可以得到较高的编码效率。和霍夫曼码相比,UVLC通用性强,解码实现较容易,但编码效率稍低。为了便于逆向解码,也可将类别码和附加码交织,如表6.8所示。表中各 a_i 构成附加码,用于传递信息,而与之交织的类别码用于提供解码时正确分断码字所需的信息。通用变长码在图像编码国际标准H.263++、H.264等标准中均有应用。

表6.6　UVLC码字结构

类别码	附加码
1	
0 1	a_0
0 0 1	a_1a_0
0 0 0 1	$a_2a_1a_0$
……	……

表6.7　UVLC码表举例

符号 x_i	码字
x_1	1
x_2	0 1 0
x_3	0 1 1
x_4	0 0 1 0 0
x_5	0 0 1 0 1
……	……

表 6.8 类别码与附加码交织的 UVLC 结构

码字结构	符号(具体码字)
1	x_1(1)
0 a_0 0	x_2(000), x_3(010)
0 a_1 1 a_0 0	x_4(00100), x_5(00110), x_6(01100), x_7(01110)
……	……

在表 6.6、表 6.7 和表 6.8 中，a_i 表示信息比特，为 0 或 1。x_i 表示待编码符号，例如表 6.8 中表明了按交织的 UVLC 编码规则，给符号 x_3 赋以码字 010 等等。

6.3.3 算术码(Arithmetic Coding)

和前述的变长码不同，算术码生成的不是分组码，即不是一个符号对应一个码字(一组比特)。在算术码中，信源符号和码字之间不存在一一对应关系。算术码是将输入的整个符号序列映射为实数轴上[0, 1)区间内的某一个子区间。消息的每个符号根据符号出现的概率减小子区间的长度。此子区间的长度等于符号序列的出现概率。当输入符号序列中的符号数目增加时，子区间长度变得更小，用于表示此子区间所需要的比特数变得更多了。因为这种技术不强制用整数个比特来表示一个信源符号，因而可更方便地趋近信源的熵。

若输入符号序列的符号表包含 N 种不同的符号 $X = \{x_1, \cdots, x_N\}$，则称为多进制算术码。当输入的符号只有二种，即 $X = \{x_0, x_1\}$，或干脆就用 0, 1 来表示的情况，称为二进制算术码(BAC)。这时输入的是由 0/1 构成的比特流，经算术码编码器编码后输出的仍为 0/1 比特流，只是在大多数情况下，比特数得到了压缩。现用一个简例来说明。

例：输入消息的符号表为 $X = \{x_0, x_1\}$，各符号出现概率为 $p_0 = 0.1$，$p_1 = 0.9$，若用霍夫曼码进行编码，因为只有二个码字，若用码字 0 来表示 x_0，码字 1 表示 x_1，则平均码长为 1 比特/符号，远大于信源熵。若用二进制算术码编码，当输入序列为 1111111××××(即头 7 个输入符号为 x_1，后面跟若干由 x_0 或 x_1 构成的尾串)的时候，BAC 的输出为 1××。即 BAC 把连续 7 个输入的符号 x_1 压缩为一个输出比特“1”，实现了高压缩比。后面将会证明算术码的编码效率非常接近于 1。在一般情况下算术码效率比霍夫曼码略高，但其抗干扰性更差。

二进制算术码在许多图像压缩标准中得到了应用(有些是作为选项)，且说明起来较方便，故以二进制算术码为例介绍算术码的工作原理。

(1) 先从概念上说明算术码怎样通过递归的码区间划分来实现输入符号串到

[0, 1)区间中某个子区间的映射的。设输入二进制符号序列为010，且已知0和1出现的概率 p_0 和 p_1。可通过图6.5所示的下述步骤实现映射：

① 在未读入输入序列前，取[0, 1)为初始码区间并可按概率 p_0、p_1 将其分为左、右二个子区间；

② 读入第一个符号。现读入的第一个符号为0，故选择与 p_0 对应的左面子区间[0, p_0)，丢掉右面的子区间；

③ 再读入第二个符号1，则按递归划分的原则，取上次取剩的[0, p_0)区间的右半，即[$p_0 \cdot p_0$, p_0)；

④ 再读入第三个符号0，取上次选中区间的左半个子区间[$p_0 \cdot p_0$, $p_0 \cdot p_0 + p_0 \cdot p_1 \cdot p_0$)，这就实现了输入符号串与码区间的映射对应关系。

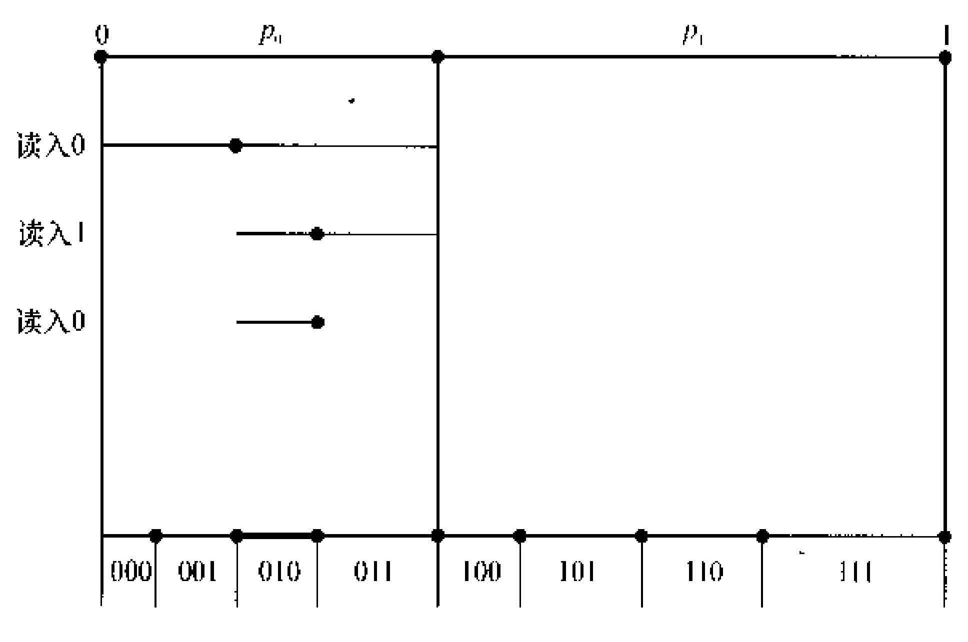


图6.5　码区间递归划分

图6.5的最末一行注明了输入不同符号串所对应的所有码区间（当输入为3个符号的消息时的情况）。码区间长度为 $p_{i_1} \cdot p_{i_2} \cdot p_{i_3}$。

上例中，若 $p_0 = p_1 = 0.5$，则上述算术码的子区间映射情况和自然二进制小数情况相一致。在 $p_0 \neq p_1$ 时，各子区间长度就不相等。且随着输入符号数的增加，码区间长度会越来越小。如果能够用某种方式表示选中了哪个子区间，且这种表示所需的平均比特数小于输入的二进制符号的长度，则实现了压缩。这里要指出两点：

一是所有与不同输入符号串对应的子区间都不重叠，且他们的总和覆盖了全部[0, 1)区间。按子区间位置与长度来推断输入符号串不会产生二义性。

二是在确定了头几个输入符号，从而对[0, 1)作了初步划分之后，再输入任何尾串，则对应的码区间总是落在已选定的码区间内部而不会跑到它外面去。

(2) 怎样表示选中的码区间的位置与长度呢？要求是既不会搞错，又能用最

少的比特数来表示。较精确的用二进制小数来表示区间的两端位置(或左端及长度)将花费太多的比特。可以在选中的码区间里尽量塞入最大的自然二进制小数对应的码区间,而用此自然二进制小数对应的比特串来表示选中的码区间。

图 6.6 说明了用 110 表示选中的码区间$[A, B)$是合适的。若少用 1 个比特,用 11 表示的范围大于$[A, B)$。若多用 1 比特,用 1100 或 1101,虽然落入$[A, B)$子区间,但浪费了 1 比特。

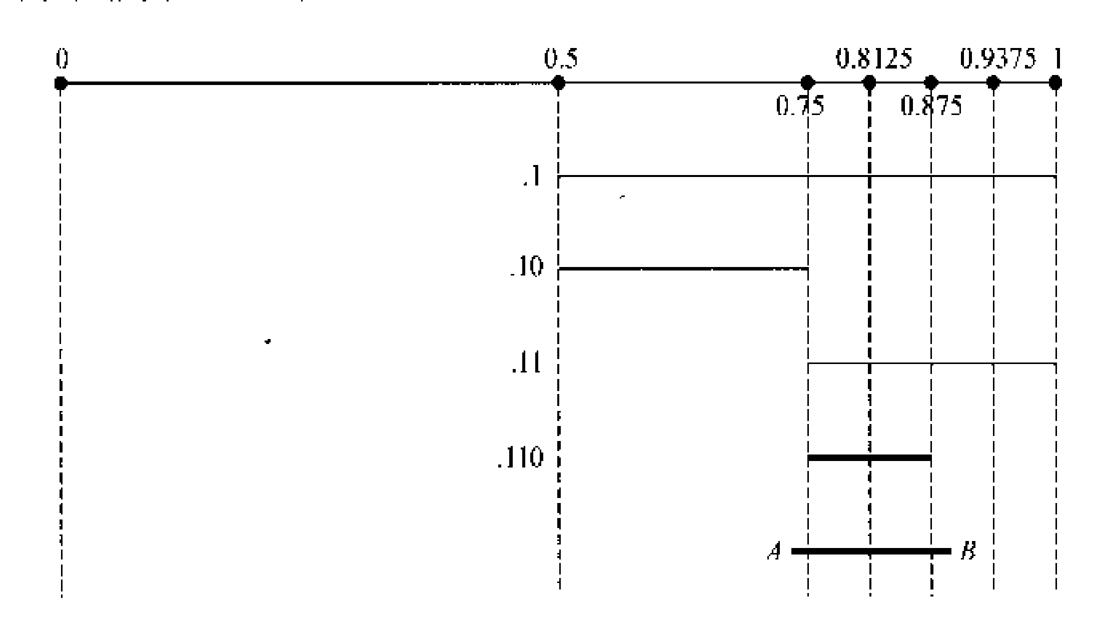


图 6.6　用二进制小数.110 表示选中的子区间$[A, B)$的示意图

(3) 二进制算术码编码步骤:

先推导码区间分割的递推公式。设在以往分析了输入的前面几个符号构成的符号串 s 之后,选定的码区间底部(左部)为 $C(s)$,宽度为 $A(s)$,如图 6.7 第一行所示。

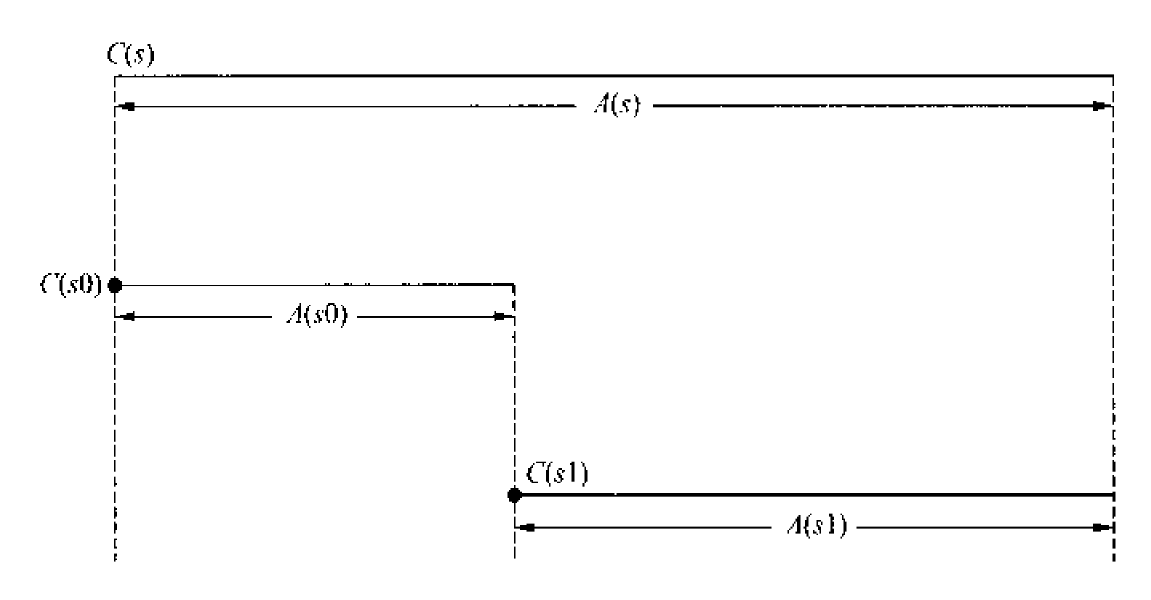


图 6.7　码区间分割递推公式符号含义图

再输入一个符号。若它为 0,则码区间底部应改为 $C(s0)$,宽度应改为 $A(s0)$

(见图6.7第二行)。若输入符号为1,则应改为$C(s1)$和$A(s1)$(见图6.7第三行)。

若输入符号0的概率为p_0,符号为1的概率为$p_1=1-p_0$,则有下列递推关系式:

当输入为0时　$$C(s0)=C(s) \tag{6.3-2a}$$

$$A(s0)=A(s)p_0 \tag{6.3-2b}$$

当输入为1时　$$C(s1)=C(s)+A(s)p_0 \tag{6.3-2c}$$

$$A(s1)=A(s)p_1 \tag{6.3-2d}$$

如果把初始状态记作以前分析的符号串为空串λ,此时$C(\lambda)=0$,$A(\lambda)=1$作为初始状态。则下面只要根据输入符号为0或1,递归地用公式(6.3-2)来修正码区间的底部$C(s)$和宽度$A(s)$,即可实现输入符号串到码区间的映射。表6.9是输入串为010,$p_0=0.4$,$p_1=0.6$时的数字实例。

由表6.9与输入字符串010对应的码区间为[.16,.256),在此区间中的最短二进制小数是.0011,即本例中算术码编码器的输出为0011,它是码区间左部$C(s)$的近似值。

表6.9　算术码编码过程表

输入	$C(s)$	$A(s)$
空串λ	$C(\lambda)=0$	$A(\lambda)=1$
0	$C(0)=C(\lambda)=0$	$A(0)=A(\lambda)\cdot p_0=0.4$
01	$C(01)=C(0)+A(0)\cdot p_0=.16$	$A(01)=A(0)\cdot p_1=.24$
010	$C(010)=C(01)=.16$	$A(010)=A(01)\cdot p_0=.096$

(4) 二进制算术码的解码步骤:解码也是递归进行的。

以第一个符号的解码来体会解码的思路。把收到的比特流S看成二进制小数,记作$C(s)$。把它和$A(0)$比,若$C(s)<A(0)$,则第一个符号必为0。因为若第一个符号为1,以后不管输入什么符号,码区间的底部必大于等于$A(0)$。反之,若$C(s)\geqslant A(0)$,则可解出第一个符号1。

递归解码步骤:

以s'表示已被解码的字符串,则判决下一个符号的过程为:

① 若$C(s)<A(s'0)$则判下一个符号$x=0$;

若$C(s)\geqslant A(s'0)$则判下一个符号$x=1$;

② 根据解出符号对$C(s)$和$A(s')$作相应修正,以"模拟"编码端的码区间分割

情况。

若在①步中判 $x=0$，则 $C(s)$ 不变，$A(s')$ 缩小为 $A(s'0)=A(s')p_0$

若判 $x=1$，则 $\quad C(s)=C(s)+A(s'0)=C(s)+A(s')p_0$

$A(s')$ 缩小为 $\qquad A(s'1)=A(s')p_1$

从初始值 $A(s')=A(\lambda)=1$ 出发，递归地反复进行①的判决解码和②的修正，可以实现解码。建议读者对本小节开始时所举例子：$p_0=0.1$，$p_1=0.9$；输入为 1111111××××，经算术码编码后输出为 1××的情况，用上述解码步骤进行解码。读者可从中体会到从输出码流中的一个比特 1 解出 7 个输入符号 1 的过程。

(5) 二进制算术码的编码效率：

有一个长度为 N 的输入符号串 $s=x_1,x_2,\cdots,x_N$，其中 $x_i=0$ 的概率为 p_0，$x_i=1$ 的概率为 $p_1=1-p_0$，符号串中共有 N_0 个 0 符号，N_1 个 1 符号，$N_0+N_1=N$，则选中的码区间长度为：

$$A(s)=p_{x_1}p_{x_2}\cdots p_{x_N}=p_0^{N_0}p_1^{N_1} \tag{6.3-3}$$

它与 s 中 0 和 1 的个数 N_0，N_1 有关，而与 0 和 1 的出现次序无关。符号 0 或 1 的出现次序只改变码区间的位置，不会改变选中的码区间的长度。

为了用最短的二进制码来表示 $A(s)$ 长度的码区间，最多需要 $-\log_2 A(s)+2$ 个比特。当 N 足够大时，N_0 近似为 Np_0，N_1 近似为 Np_1。于是由 $A(s)$ 长度表达式(6.3-3)，最多需

$$-Np_0\log_2 p_0-Np_1\log_2 p_1+2=NH(x)+2$$

个比特。故平均码长

$$L=H(x)+\frac{2}{N}$$

当 N 很大时平均码长趋近于信源的熵。

本节只是介绍了二进制算术码的原理，对于含 n 种符号的多进制算术码，则要把码区间按各符号出现概率划分为 n 个子区间，选中与输入的第一个符号相对应的子区间，如此递归的划分、选择即可。要注意的是，需引入某种机制来指明算术码码流的结束。例如：专门引入一个字符串"结束符"，它只在输入字符串结束时使用一次。解码端解出此"结束符"表明整个编解码过程的结束。

在算术码实际实现中，要解决许多实际问题，例如计算精度问题、符号出现概率的估计问题等。

随输入符号串变长，码区间越来越小，必然发生运算精度问题。这时可以将码区间放大，例如在必要时左移 1 位(乘 2)。有许多种实现方法巧妙地解决

各种实现中的疑难问题。希望本小节的介绍有助于读者在阅读相关文献时理解它们。

6.4 预 测 编 码

在图像编码一般框图中映射变换模块所做的工作是对待编码图像进行预测，并将预测差输出，供量化编码，而在接收端将经量化的预测差与预测值相加以恢复原图，则这种编码方法就称为预测编码。

因为是对预测差进行编码，故也可称为"差值脉冲编码调制(DPCM)"。在许多图像编码国际标准中都采用了预测编码技术。

若收发两端的预测器完全相同，且发端不经过量化，那么在接收端完全可以实现无失真地恢复原图像。即预测本身并不引入误差。然而在实际应用中常常使用量化器以提高压缩比，从而使恢复的图像含有失真。

6.4.1 预测编码原理

为便于分析，把像素按某种次序排成一维序列(如按电视扫描次序)$\{x_n \mid n=1, 2, \cdots\}$，$x_n$ 表示某个像素的灰度值。根据条件熵理论，若从已知的 $x_{n-1}, \cdots, x_{n-m}$ 来估计 x_n(m 点估计)，则由于相邻像素间有较大相关性，可以较准确地估计出 x_n，即估计值 $\hat{x}_n = f(x_{n-1}, \cdots, x_{n-m})$ 近似等于 x_n，或预测误差 $e_n = x_n - \hat{x}_n$ 的方差 $\sigma_{e_n}^2$ 较小，并小于原数据的方差 $\sigma_{x_n}^2$。由(6.2-8)式，对预测差进行编码所需的比特数要比直接对原数据编码来得少，从而实现了压缩。比特率大致可减少 $\frac{1}{2}\log_2 \frac{\sigma_{x_n}^2}{\sigma_{e_n}^2}$ bit/pel 左右。图6.8是预测编码的原理框图。

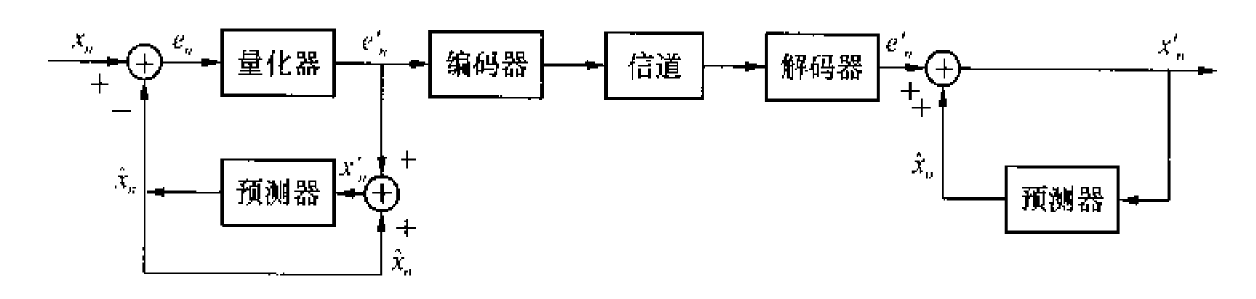


图6.8 预测编码框图

由框图看出，这是一个带负反馈的非线性系统，要对它进行严格的分析相当困难。常用的分析方法是对预测器和量化器分别进行讨论，这样做得到的虽只是局

部最优解,但可简化分析。要记住预测器和量化器的设计不是完全独立的。它们对环路稳定性有影响,它们的设计和信源的统计特性有很大的关系。预测器愈好,能使预测差的方差愈小,则压缩比就愈高。因而预测器设计是 DPCM 的核心问题。预测器可以是固定的,也可以是自适应的。它可以是线性的,也可以是非线性的。在下一小节中将对此作详细的讨论。

1. 预测编码框图的工作过程

(1) x_n 与发端预测器产生的预测值$\hat{x}_n$ 相减得预测差 e_n。

(2) e_n 经量化后变为 e_n',同时引入量化误差。

(3) e_n'再经过编码器编成码字(如 Huffman 码)发送,同时编码端又将 e_n'加上$\hat{x}_n$ 得到 x_n'。因存在量化误差,$x_n'\neq x_n$,但相当接近。发端的预测器及其环路作为发端本地解码器。

(4) 发端预测器带有存储器,它把 x_{n-1}', …, x_{n-m}'存储起来以供对 x_n 进行预测得到$\hat{x}_n$ 之用。

(5) 继续输入下一个像素,即 $n=n+1$,重复上述过程。

解码端的工作过程和编码端中本地解码器完全一样。若不存在传输误码,则解码器输出为 x_n'。

2. 要点分析

(1) 发端必须有本地解码器,它为预测器对 x_n 进行预测创造条件。

(2) 收端解码器必须和发端本地解码器完全一样,即必须保证收发的预测方法、预测条件完全一致。

(3) 具体实现时是由 x_{n-1}', …, x_{n-m}'来预测 x_n 的,这是因为接收端只能得到 x_{n-1}',故发端必须使用 x_{n-1}'来作预测,以确保收发二端"同步"运行。否则会产生"漂移"现象,整个系统就不能正常工作。

下面用一个数字实例来对漂移现象作形象的描述。

例如,$\{x_n\}=\{0, 1, 2, 3, 4, \cdots\}$,量化器设计为:$[-1, 0, 1]\to 0$、$[2, 3, 4]\to 3$ 等,设 $\hat{x}_1=0$。

① 若发端把 x_{n-1} 作为 x_n 的预测值,而不是用 x_{n-1}' 作为 x_n 的预测值,则会产生漂移现象;

现在 $e_n=\{1, 1, 1, \cdots\}$,经过量化后,$e_n'=\{0, 0, 0, \cdots\}$,所以 $x_n'=\hat{x}_n+e_n'=\{0, 0, 0, \cdots\}$,即收端复原值恒为 0,它不随 x_n 的缓慢增大而增大。即 x_n'与 x_n 的误差会越来越大。这就是所谓的漂移现象。

② 若发端把 x_{n-1}' 作为 x_n 的预测值,则 x_n'始终能跟上 x_n 的变化,没有漂移,只有量化误差。

预测编码的工作过程中各参数的值如表6.10所示。

表6.10 DPCM工作过程中各参数取值

x_n	0	1	2	3	4	5	6	7	…
$\hat{x}_n$	0	0	0	3	3	3	6	6	…
e_n	0	1	2	0	1	2	0	1	…
e_n'	0	0	3	0	0	3	0	0	…
$\hat{x}_n+e_n'=x_n'$	0	0	3	3	3	6	6	6	…

从上表可以看出，收端复原值为$x_n'=\{0, 0, 3, 3, 3, 6, 6, 6, \cdots\}$，虽然与发端的输入之间存在误差，但能"跟上"$x_n$的变化，这种误差是由于量化器的使用而引起的，没有漂移现象。

本例形象地说明了预测器必须以含有量化误差的x_n'作为参考进行预测的必要性。下面转到对预测器的分析。

6.4.2 线性预测

在上小节中已强调过根据x_{n-i}'而不是x_{n-i}进行预测的必要性。在本小节中为了表示的方便，用x_{n-i}来表示框图中的x_{n-i}'以简化公式的书写。

1. 一维情况

为了便于说明问题，先从一维谈起。如果$\hat{x}_n$是$x_{n-1}, \cdots, x_{n-m}$的线性组合，则称为线性预测，即

$$\hat{x}_n \equiv \hat{x}(n) = \sum_{k=1}^{m} a_k x(n-k) \tag{6.4-1}$$

$$e_n \equiv e(n) = x_n - \hat{x}(n) = x(n) - \sum_{k=1}^{m} a_k x(n-k) \tag{6.4-2}$$

这里a_k为预测参数，m为预测阶数。求最优线性预测就是求a_k，使$\sigma_{e_n}^2$为最小。设$x(n)$是广义平稳的随机过程，不失一般性，可以设$E\{x(n)\}=0$，则

$$\sigma_{e_n}^2 = E\{e_n^2\} = E\left\{\left[x(n) - \sum_{k=1}^{m} a_k x(n-k)\right]^2\right\} \tag{6.4-3}$$

要选a_k使$\sigma_{e_n}^2$最小，只要令$\dfrac{\partial \sigma_{e_n}^2}{\partial a_i}=0$，$(i=1, 2, \cdots, m)$，联立解出$a_k$，即

$$\frac{\partial \sigma_{e_n}^2}{\partial a_i} = -2E\left\{x(n-i)\left[x(n) - \sum_{k=1}^{m} a_k x(n-k)\right]\right\} = 0 \tag{6.4-4}$$

这里 $i=1,2,\cdots,m$。因为 $x(n)$ 的自相关函数

$$R(k)=E\{x(n)x(n-k)\} \quad 且有\ R(-k)=R(k)$$

代入式(6.4-4)得

$$R(i)-\sum_{k=1}^{m}a_kR(|k-i|)=0 \qquad i=1,2,\cdots,m \tag{6.4-5}$$

为了求解 a_k，先要估计 $x(n)$ 的自相关函数 $R(k)$。设 $x(n)$ 是各态历经的，而且有足够多的样本 N，则

$$R(k)\cong\frac{1}{N}\sum_{n=1}^{N}x(n)x(n-k)$$

于是式(6.4-5)可写成矩阵形式

$$\begin{bmatrix} R(0) & R(1) & \cdots & R(M-1) \\ R(1) & R(0) & \cdots & R(M-2) \\ \cdots & \cdots & \cdots & \cdots \\ R(M-1) & R(M-2) & \cdots & R(0) \end{bmatrix}\begin{bmatrix} a_1 \\ \cdots \\ \cdots \\ a_m \end{bmatrix}=\begin{bmatrix} R(1) \\ R(2) \\ \cdots \\ R(M) \end{bmatrix} \tag{6.4-6}$$

上式左边的矩阵是 $x(n)$ 的相关矩阵，是个 Toeplitz 矩阵，可以用 Levinson 算法等方法解出 a_k。这样求得的线性预测器是在均方误差最小意义上的最优线性预测。下面进一步研究其性质，我们先来证明

$$\sigma_{e_n}^2=R(0)-\sum_{k=1}^{m}a_kR(k) \tag{6.4-7}$$

证：由式(6.4-4)，从 $\frac{\partial\sigma_{e_n}^2}{\partial a_i}=0$ 解得的 a_i 应能满足

$$E\left\{x(n-i)\left[x(n)-\sum_{k=1}^{m}a_kx(n-k)\right]\right\}=0$$

上式两边同乘 a_i 并对 i 求和，由于 $\hat{x}_n$ 是线性运算，$E\{\cdot\}$ 也是线性运算，故可得

$$E\{\hat{x}(n)[x(n)-\hat{x}(n)]\}=0 \tag{6.4-8}$$

由式(6.4-8)，考虑到 $E\{x(n)\}=0$，有

$$\begin{aligned}\sigma_{e_n}^2&=E\{e^2(n)\}=E\{[x(n)-\hat{x}(n)]^2\}=E\{x(n)[x(n)-\hat{x}(n)]\}\\&=R(0)-\sum_{k=1}^{m}a_kR(k)\end{aligned}$$

证毕。

由式(6.4-7)可以看到，e_n 的方差小于$R(0)$，现在$E\{x(n)\}=0$，故e_n 的方差小于 $x(n)$的方差。所以传送差值 e_n 比直接传送原始信号 $x(n)$有利于压缩。$x(n)$的自相关性愈大，即$R(k)$愈大，差值 e_n 的方差就愈小，所能达到的压缩比就愈大。反之，若 $x(n)$前后符号间互不相关，$R(k)=0$（当$k\neq 0$）这时$\sigma_{e_n}^2=R(0)-\sigma_{\hat{x}_n}^2$，则不能压缩。

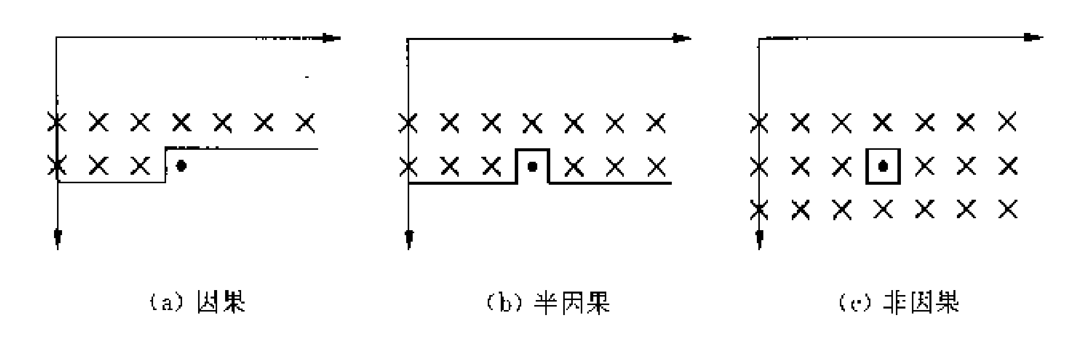
关于预测阶数 m 的选择。一般说来，刚开始时随着 m 值的增大$\sigma_{e_n}^2$将减小。可以证明当 m 足够大，即到了能使$E\{e(n)e(n+j)\}=0\ (j\neq 0)$ 的时候，再增加 m 值并不会使 $\sigma_{e_n}^2$ 再变小。一般来说，若 $x(n)$为平稳的 m 阶马尔可夫过程，则 m 阶最佳线性预测器就是在 $\sigma_{e_n}^2$ 最小意义上的最好预测器。

2. 二维情况

若原始图像用 $f(m, n)$来表示，它的二维线性预测为

$$\hat{f}(m, n)=\sum_{(k, l)}\sum_{\in Z} a_{kl} f(m-k, n-l) \qquad (6.4-9)$$

式中：a_{kl}为二维预测系数；Z 为预测区域；$(k, l)\in Z$ 表示用 Z 中的点来对 $f(m, n)$进行预测。显然，Z 应为(m, n)点的邻域，但不包括(m, n)点本身。可以用电视扫描作为因果性的依据把 Z 分为三类，如图 6.9 所示。图中(a)为因果性区域，(b)为半因果性区域，(c)为非因果性区域。最常用的是情况(a)。



(a) 因果　　(b) 半因果　　(c) 非因果

图 6.9　预测区域示意图

设 $f(m, n)$为二维平稳随机过程，且 $E\{f(m, n)\}=0$。则 $x(n)$的自相关函数 $R(k, l)$也是 $f(m, n)$的协方差函数。且$R(k, l)=R(\pm k, \pm l)$。预测的差值

$$e(m, n)=f(m, n)-\hat{f}(m, n)$$

因为 $E\{e(m, n)\}=0$，故有

$$\sigma_{e_{mn}}^2=E\{e^2(m, n)\}=E\Big\{\Big[f(m, n)-\sum_{(k, l)}\sum_{\in Z} a_{kl} f(m-k, n-l)\Big]^2\Big\}$$

和一维情况下的做法相类似，求出$\partial\sigma_{e_{mn}}^2/\partial a_{ij}$，令其为 0，得

$$E\left\{f(m-i,\ n-j)\left[f(m,\ n)-\sum_{(k,\ l)}\sum_{\in Z}a_{kl}f(m-k,\ n-l)\right]\right\}=0$$

(6.4-10)

即
$$R(i,\ j)-\sum_{(k,\ l)}\sum_{\in Z}a_{kl}R(k-i,\ l-j)=0$$

用堆叠求解法，即把下标$(k,\ l)$、$(i,\ j)$堆叠为一维下标再求解，可以求出最佳系数$\{a_{kl},\ (k,\ l)\in Z\}$。

6.4.3 运动补偿和帧间预测

采用帧间预测编码可以减少时间域上的冗余度，提高压缩比。如果将上一帧相同空间位置处的像素值作为待编码的当前帧的像素预测值，这种预测对图像中的静止部分将是很有效的。但是对于运动部分，这种不考虑物体运动的简单的帧间预测效果并不好。如果有办法在对当前帧某像素(或像素块)进行预测时知道这个像素(或像素块)是上一帧的哪个位置移动过来的，在做预测时以那个位置上的像素值作为预测值，则预测的准确性将大为提高。也就是说采用运动补偿帧间预测可以更好地利用序列图像的时间冗余度使预测差的方差大大减少，从而减低码率，提高压缩比。这项技术已广泛应用于活动图像编码的国际标准之中，取得了很好的效果。显然，获得好的运动补偿的关键是运动估计。本小节讨论怎样从图像序列中获得有足够精度的运动矢量，并据此进行运动补偿的技术。在这里，我们仅讨论图像序列的二维运动估计问题。

运动估计技术主要分成三大类：像素递归法、块匹配法和相位相关法，目前最常用的方法是块匹配法。

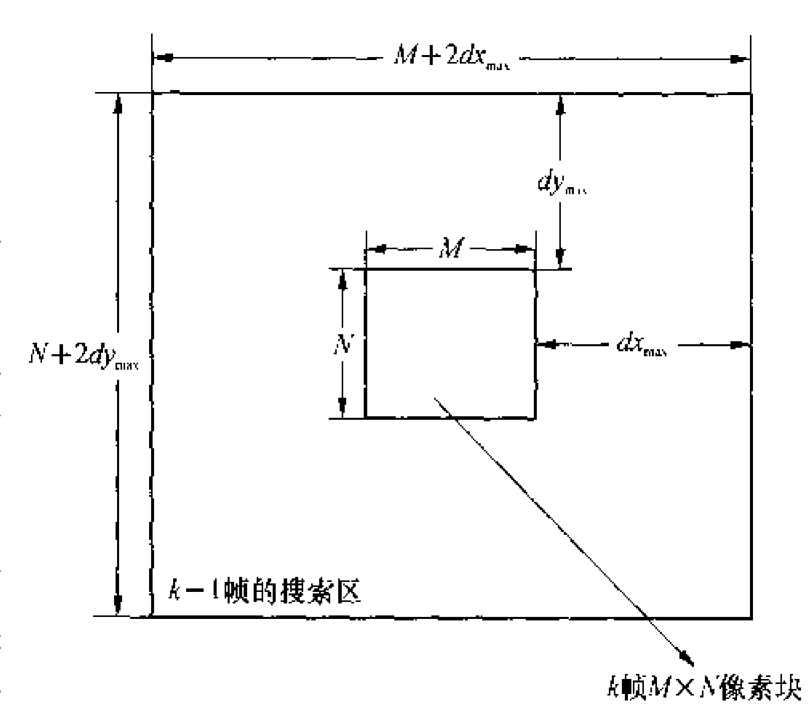


图 6.10 待匹配像素块与搜索区的几何位置关系

将图像分割成$M\times N$的小块，假设块内各像素做相同的运动，且只做平移运动。虽然实际上块内各点运动不一定相同，也不一定只有平移运动，但当$M\times N$较小时，上述假设可近似成立。这样做的目的只是为了简化运算。块匹配法对当前帧图像的每一块，在上一帧的一定范围内搜

索最优匹配，并认为本块就是从上一帧最优匹配块位置处平移过来的。设可能的最大偏移矢量为 dx_{max}，dy_{max}，则搜索范围为 $(M+2\times dx_{max})\times(N+2\times dy_{max})$。图 6.10 示出了待匹配块与搜索区的几何位置关系。

1. 最优匹配准则

目前，有三种常用的匹配准则，它们是最小平均绝对差(MAD)、最小均方误差(MSE)和归一化互相关函数(NCCF)。分别定义如下：

(1) 最小平均绝对差：

$$\mathrm{MAD}(i,\ j)=\frac{1}{MN}\sum_{m=1}^{M}\sum_{n=1}^{N}|f_k(m,\ n)-f_{k-1}(m+i,\ n+j)| \tag{6.4-11}$$

式中，$(i,\ j)$为位移矢量。若在某一个$(i,\ j)$处 $\mathrm{MAD}(i,\ j)$达到最小，则该点就是要找的最优匹配点。这里 f_k 和 f_{k-1} 分别为当前帧和上一帧的灰度值。

(2) 最小均方误差：

$$\mathrm{MSE}(i,\ j)=\frac{1}{MN}\sum_{m=1}^{M}\sum_{n=1}^{N}[f_k(m,\ n)-f_{k-1}(m+i,\ n+j)]^2 \tag{6.4-12}$$

同样是以 MSE 值最小为最优匹配点。

(3) 归一化互相关函数：

$$\mathrm{NCCF}(i,\ j)=\frac{\sum_{m=1}^{M}\sum_{n=1}^{N}f_k(m,\ n)f_{k-1}(m+i,\ n+j)}{\left[\sum_{m=1}^{M}\sum_{n=1}^{N}f_k^2(m,\ n)\right]^{\frac{1}{2}}\left[\sum_{m=1}^{M}\sum_{n=1}^{N}f_{k-1}^2(m,\ n)\right]^{\frac{1}{2}}} \tag{6.4-13}$$

这里，最优匹配相应于找 NCCF 的最大点。

上述三准则中，由于 MAD 准则不需作乘法，实现简单、方便，所以使用得最多。

2. 搜索方法

有了匹配准则，剩下的问题就是寻找最优匹配点的搜索方法。最简单、可靠的方法是穷尽搜索。即对搜索窗内所有可能的候选位置都计算 $\mathrm{MAD}(i,\ j)$值，从中找出最小 MAD，其对应的偏移量即为所求的运动矢量。此法计算量非常大，但总可找到全局最优点。为减小计算量，提出了许多运动估计快速算法，这些算法计算量小，但往往难以保证找到的是全局最优解。具体方法有一次一个方向搜索法、二维对数法、三步法、新三步法等等，可参阅相关文献。

通过运动估计得到运动矢量之后，即可根据运动矢量从参考帧（在本小节中即为上一帧）中相应位置取出像素块来，作为本帧当前块的预测值，从而实现了运动补偿帧间预测。

3. 三点说明

（1）图像块大小的选择和块内运动不一致问题。

块的大小的选择受两个矛盾因素的约束。块大时，块内各像素作相同大小和方向的平移运动这个假设条件不容易得到满足，从而影响了预测的准确性。但块太小，则运动估计结果易受噪声的影响，估计值不够可靠，而且块小了所需传输的运动矢量个数增加，这也不利于压缩比的提高。应做到二者兼顾。早期的活动图像编码标准如 H.261、MPEG-1 和 MPEG-2 都采用 16×16 像素（亮度分量）大小的块作为运动补偿帧间预测的单元。在图像分辨率较低的情况下，块内运动不一致性显得更加突出，这时可以把 16×16 像素块作进一步地划分，把 8×8 甚至 4×4等大小的块作为运动估计和运动补偿的单元，来提高帧间预测精度（参见 H.263、MPEG-4、H.264 等国际标准）。

（2）对运动矢量精度的要求。

图像中物体在帧间的真实位移不一定是取样网格的整倍数。因此，整像素运动矢量精度往往显得不够。这时要求 1/2 像素、1/4 像素甚至更高的精度。为了实现亚像元精度的运动估计和运动补偿，需要对参考帧图像进行相应的空间内插，并要求用较好的内插方法来保证内插精度从而实现较精确的亚像元精度的运动补偿帧间预测。

（3）参考帧的获得。

根据以前强调的预测编码原理，作运动补偿时，参考帧必须是由编码端的本地解码器恢复的，含有量化误差的上一帧（用 x_n' 或 f_{k-1}' 来表示），而不是上一个输入帧（x_n 或 f_{k-1}）。注意，在解码端也能得到与之完全相同的参考帧 f_{k-1}'，但在解码端不可能得到 f_{k-1}。

至于运动估计时，为得到运动矢量而进行搜索时，所用的“上一帧”参考帧既可以是 f_{k-1}' 也可以是 f_{k-1}。我们把为什么可以这样做的原因留作习题供读者思考。注意运动矢量是通过码流传送到接收端的，解码器不必通过搜索来找到运动矢量而只需在码流中相应位置获取运动矢量。

6.5 变换编码

在图 6.1 的压缩编码框图中，若用某种形式的正交变换来实现框图中的映射

变换，这种编码方式就称为变换编码。图6.11是变换编码系统的编解码框图。在发端，一般是将原图 $f(m, n)$ 分成子像块，对每个子像块进行某种形式的正交变换，变成变换域中的系数阵列 $F(s, t)$，系数选择器选择 $F(s, t)$ 中的若干主要的分量，经量化和编码变为数码进行传输。在收端经熵解码器和反量化器将收到的数码解码成 $F'(s, t)$。由于在发端为了压缩数据量而进行了系数选择，在编码时对所选择的系数又进行了量化，因此在收端所恢复的 $F'(s, t)$ 是有一定的失真的。它经过反变换后得到的复原图像 $f'(m, n)$ 也是有一定失真的。如果系数选择器和量化器设计得好，可以把这种失真限制在允许的范围内。因此说，变换编码是一种限失真编码。本节要讨论的中心问题是怎样设计系数选择器和量化器，使得能够在低传输速率下使失真限制在允许的限度之内。要注意框图中的反量化器不能消除量化误差，只起尺度转换作用。正交变换的方式对变换编码的性能也有影响，这也属于本节的讨论范围。至于各种正交变换的定义、性质，快速算法等内容已在第三章中详细讨论了，在此不再重复。

为什么在编码前先进行正交变换可以压缩数据量呢？第三章详细讨论过变换域中的能量集中问题，指出一般变换域的能量总比空间域集中。例如，图像信号的能量主要集中在低频部分，如果在编码时略去某些能量很小的高频分量，或在量化时对方差较小的分量分配较少的比特数，这样可以明显降低比特率。这就是变换压缩的基本思想。

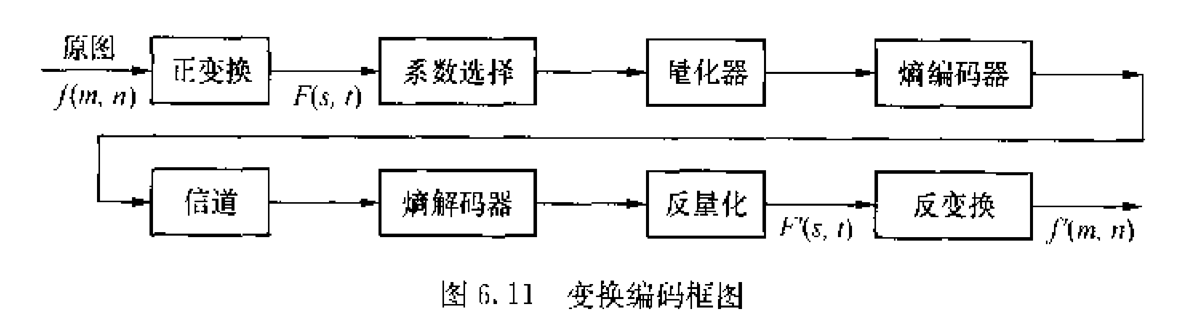


图6.11　变换编码框图

6.5.1　变换方式及子像块大小

第三章对图像的正交变换已经作了详细的讨论，这里不再重复。由第三章图3.16及有关分析可以看出，若图像是广义平稳的，经过卡-洛变换后的各系数互不相关，能量最为集中。由于舍弃数值较小的系数（不进行编码传输），所造成的均方误差最小，见(3.6-19)式，故卡-洛变换是均方误差最小意义下的最优变换。它可作为研究各种变换方式对压缩比影响的比较基准。但是，由于卡-洛变换的基向量是原始子图像协方差阵的特征向量，它不是固定方向的，而是随着待编码子图像的内容而改变的，因此实现起来比较困难。再者，很多图像不满足平稳性的假定，加之均方误差也不是一个很好的质量评价标准，故卡-洛变换的实用价值受到限制。

事实上对实际图像的变换编码来说,以图像是不平稳随机过程的假设为前提的自适应变换较卡-洛变换更为有效。第三章介绍过不少固定基向量的正交变换,它们的压缩性能与卡-洛变换差别不很大,其中尤其以离散余弦变换(DCT)与卡-洛变换的性能最为接近。但是DCT是固定基向量的,易于实现,因此用得较多。

变换编码时由于只能利用小块内的相关性来压缩图像,故块若选得太小,不利于压缩比的提高。而若块太大,则不但计算复杂,而且距离较远的像素间相关性不大,对提高压缩比的效果不明显。目前一般选8×8块为最常见,也有用16×16、4×4块作变换的。

6.5.2 系数选择和比特分配

选择哪些变换系数进行量化编码,对选中的系数以怎样的精度来近似表示,对比特率和失真有很大影响。因而本小节是提高变换编码性能的关键。

1. 系数选择

系数选择通常有两种方法:区域编码和阈值编码。

(1) 区域编码就是指只对指定形状区域里的变换系数进行量化编码,丢弃区域外的系数。从率失真理论的观点,应选取方差大的系数。即:若将$M\times N$的图像分成$n\times n$大小的子块,共有$\frac{MN}{n^2}$个子块。每块变换后有n^2个系数$F_k(s, t)$。这里s、t为空间频率,取值范围为0, …, $n-1$。k是子块编号,取值范围为0, …, $\frac{MN}{n^2}-1$。对每个空间频率(s, t)的系数,可以由$\frac{MN}{n^2}$个值估计出它的方差来。于是可在这n^2个系数中选取方差最大的若干个系数作进一步地量化、编码、传输。一般选中的系数为直流和低频分量,对DCT变换而言,集中在变换系数矩阵的左上角,这是由于变换系数的能量集中效应。变换系数在选择区域形状时除了依据变换系数方差大小外,也可结合视觉特性,把人眼较敏感的空间频率对应的系数纳入选中的区域内。当图像编码前经过预滤波时,编码系数区域形状与预滤波通带会较接近。选中的系数越少,比特率就越低,当然失真会越大。总之,根据对码率的要求合理选择方差大、视觉敏感的系数是要点,这些系数一般在变换系数矩阵的左上角。采用区域编码时所有子块的编码系数区域都是相同的,没有自适应能力,对于空间细节丰富的块,因系数选择不恰当引起的失真较大。这种失真往往体现为因高频分量丢失引起的图像模糊。

(2) 阈值编码则是选择幅值超过某个设定阈值的系数,对它们进行量化编码,舍弃幅值小于阈值的系数。它有一定的自适应能力,因而可得到较好的复原图像质量。只是因为超过阈值系数的位置对每子块是不同的,因而在编码时不仅要对

变换系数编码，也要对这些系数在变换系数矩阵中的位置进行编码。为减小对系数位置编码所需的比特数，要采用一定的技巧。最常用的方法是先用之字形扫描将系数矩阵拉成一维，取阈值后用行程码对零行程进行编码。在不同的国际标准中，对未被选中系数的零行程和选中系数经量化后的非零值怎样结合起来编码有各种不同的方法，可参见这些标准的文本。图 6.12 是 8×8 变换系数块拉成一维时的扫描次序示意图，它被广泛应用于各种图像编码国际标准中，在某些情况下也可采用其他扫描次序。

0	1	5	6	14	15	27	28
2	4	7	13	16	26	29	42
3	8	12	17	25	30	41	43
9	11	18	24	31	40	44	53
10	19	23	32	39	45	52	54
20	22	33	38	46	51	55	60
21	34	37	47	50	56	59	61
35	36	48	49	57	58	62	63

图 6.12 系数矩阵之字形扫描次序示意图

2. 比特分配

上面解决了选哪些系数进行量化编码的问题，现在要解决对选中的系数以什么样的精度来表示的问题。这和量化器形状和量化步长有紧密关系。虽然有许多文献介绍了不同形状量化器的设计方法，在实用中，许多图像编码国际标准都采用线性量化器，即

$$F_q(s, t) = \text{round}\left(\frac{F(s, t)}{q}\right)$$

式中：q 为量化步长。此量化曲线在 $\pm 0.5q$ 的过零区间有一个死区，在此区间 $F_q(s, t)$ 等于零。在 H.264 标准中允许对死区的宽度有少量调整，使之大于 1 个量化步长。

由于人眼对不同空间频率有不同的敏感程度，在选量化步长时，应该对人眼敏感的系数选较小量化步长，而对视觉不敏感系数可以粗量化。但对 n^2 个不同空间频率的系数分别用 n^2 个不同的量化步长比较麻烦，常用的做法是设定一个量化加权矩阵 $Q(s, t)$。用 $Q(s, t)$ 对 $F(s, t)$ 归一化后用统一的量化步长进行量化，则操作较方便。

$$F_q(s, t) = \text{round}\left(\frac{F(s, t)}{q \cdot Q(s, t)}\right)$$

式中 q 的大小确定了所选阈值，q 大则量化粗糙、比特率低、失真大。可以根据对比特率和失真的要求来选择 q 的大小。

图 6.13 是 JPEG 默认的量化加权矩阵，图中(a)为亮度分量的量化加权矩阵，(b)是色差分量的量化加权矩阵 $Q(s, t)$。

16	11	10	16	24	40	51	61
12	12	14	19	26	58	60	55
14	13	16	24	40	57	69	56
14	17	22	29	51	87	80	62
18	22	37	56	68	109	103	77
24	35	55	64	81	104	113	92
49	64	78	87	103	121	120	101
72	92	95	98	112	100	103	99

(a)

17	18	24	47	66	99	99	99
18	21	26	66	99	99	99	99
24	26	56	99	99	99	99	99
47	66	99	99	99	99	99	99
99	99	99	99	99	99	99	99
99	99	99	99	99	99	99	99
99	99	99	99	99	99	99	99
99	99	99	99	99	99	99	99

(b)

图 6.13 JPEG 默认的量化加权矩阵

6.5.3 方块效应的消除

变换编码引入的失真大致有三种类型：

(1) 分辨率下降。这是因为忽略了部分高频分量及粗糙量化高频分量的结果。

(2) 在图像灰度平坦区有颗粒噪声出现。这种失真常常是由那些只分到 1 bit 的高频分量系数带来的。由于只有 1 bit，故无法精确描述这些系数，引入了较大的量化噪声。同时，在平坦区，视觉阈值较低。

(3) 方块效应。这是变换编码中最令人头痛的失真，因为人眼对此非常敏感，克服方块效应有两种办法：

① 交叠分块法。在将图像分块时，块的划分使块与块之间有交叠部分，如一个像素重叠。解码复原图像后，再对块边缘进行平均。这样做的代价是由于块增多而使编码效率有所下降。

② 后滤波法。解码后作低通处理，将块边界处的"突跳"滤平。代价是图像细节也减少了。为了减轻这种细节的丢失，可采用自适应的块效应滤波器，即当方块的边界像素不处于图像边缘时进行平滑滤波，以消除块效应的影响，而当方块的边界像素处于图像中物体边缘时就不对它作平滑滤波，以减轻图像边缘变模糊的效应。在 H.264 等国际标准中有简单有效的去块效应自适应滤波算法的详细介绍。

6.6 图像编码国际标准简介

十余年来，图像编码技术日臻成熟，得到了越来越广泛的应用，它已成为广播、

通信、计算机三大技术融合的技术原动力之一。反映图像编码最新技术成果,适合各种应用的国际标准的推出既是技术和应用发展的需要,同时标准的推出也大大推进了数字电视、图像通信技术的产业化和推广的进程,使人们的生活方式随之发生改变。MPEG-1的制订和VCD在中国的普及就是一个证明。而MPEG-2和MPEG-4的制订为DVD、数字电视广播、流媒体和网络电视的发展和推广奠定了技术基础。本节对近年来制订的图像编码国际标准的概况作简述。

表6.11列出近年来制订的一些图像编码国际标准。表中列出了三种类型图像的编码标准,它们是:二值图像、连续调静止图像和活动图像的编码标准。所有活动图像编码标准全都采用了混合编码技术,本节中以H.261标准为例对混合编码作简要说明,后来的标准在此基础上又有新的发展和改进。

表6.11 图像编码国际标准

简 称	标准编号	推出日期	标准及说明
H.261	ITU-T H.261	1990年	P×64 kbps视听业务的视频编解码
H.263	ITU-T H.263	1996年	低比特率通信的视频编码
H.263+	ITU-T H.263 Ver.2	1998年	H.263第二版
H.263++	ITU-T H.263 Ver.3	2000年	H.263第三版
H.264	ITU-T H.264 ISO/IEC 14496 Part10	2003年	先进视频编码(AVC)
JPEG	ISO/IEC 10918-1	1992年	连续调静止图像压缩编码
JPEG-LS	ISO/IEC 14495-1	1999年	连续调静止图像无损/近于无损压缩编码
JPEG2000	ISO/IEC 15444	2000年	新的连续调静止图像压缩编码标准
MPEG-1	ISO/IEC 11172	1993年	用于数字存储的活动图像和声音的编码
MPEG-2	ISO/IEC 13818	1994年	活动图像和声音的通用编码
MPEG-4	ISO/IEC 14496	1999年	音视频对象的通用编码
JBIG1	ISO/IEC 11544	1993年	二值图像编码标准(第一版)
JBIG2	ISO/IEC 14492	1999年	二值图像编码标准(第二版)

6.6.1 H.26×系列视频编码标准

1. H.261

H.261是CCITT(即现在的ITU-T)第15研究组(SGXV)为在ISDN信道上

进行会议电视和可视电话业务而制订的视频编码标准。其速率为 P * 64 kbps，P 从 1 至 30 即 64 kbps 至 1.92 Mbps。

自 1990 年 H.261 标准问世以来，以运动补偿帧间预测和分块 DCT 相结合的混合编码方案已成为视频编码（即活动图像编码或序列图像编码）的经典方案。该方案用运动补偿帧间预测消除时间冗余度，对预测差进行 DCT 变换编码以消除空间冗余度，再用量化和熵编码器来消除视觉冗余度和符号编码冗余度，实现了低失真高压缩比的视频编码。下面以 H.261 为例介绍混合编码的原理。

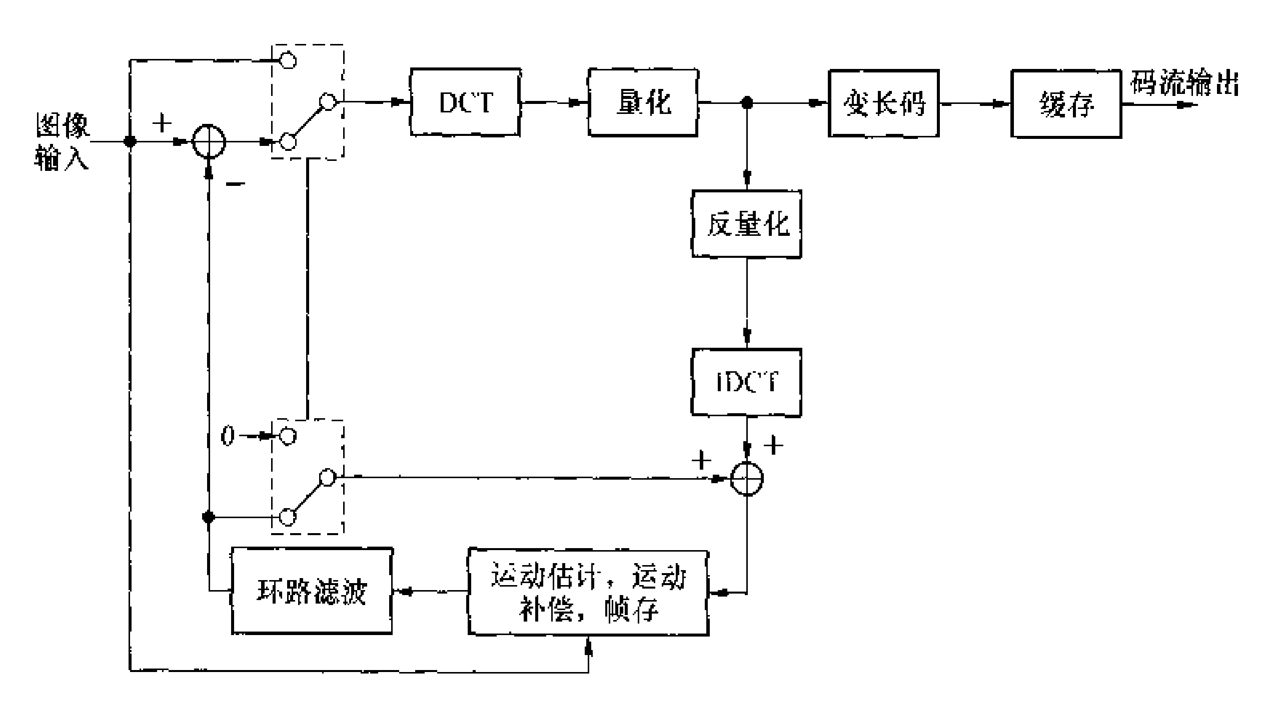


图 6.14　H.261 建议的视频编码器

图 6.14 为 H.261 视频编码器原理框图。其工作过程如下：当前帧输入图像 $[f_k(m, n)]$ 和它的运动补偿帧间预测图 $[\hat{f}_k(m, n)]$ 相减，得预测差。对预测差进行分块 DCT 变换得变换系数 $F(s, t)$，对 $F(s, t)$ 进行量化得 $F_q(s, t)$，然后编码成码流输出。编码端的本地解码器对 $F_q(s, t)$ 进行反量化、反 DCT，然后与预测图 $[\hat{f}_k(m, n)]$ 相加，得到复原图 $[f_k'(m, n)]$，存入帧存，以便在对后面帧进行编码时作为预测时的参考帧。图中“运动估计，运动补偿、帧存”这个功能模块包含运动估计(ME)、运动补偿(MC)和帧存三项功能。由运动估计得到的运动矢量 MV 在帧存中合适位置找到当前输入块的帧间预测 $[\hat{f}_k(m, n)]$。

在 H.261 中运动估计是以 16×16 亮度块为单位用块匹配法进行的，DCT 以 8×8 为单位进行，图中的二个开关符号表示对当前宏块作帧内编码还是帧间预测编码的选择。量化编码后的变换系数要与其他许多附加信息按规定的格式复用成为一个码流输出，这些细节不在此赘述，有兴趣的读者可参阅 H.261 标准文本。关于混合编码框图只作下列补充说明：

(1) 框图中编码模式选择与多个因素有关。对当前宏块选择帧内还是帧间编码的主要依据是运动补偿帧间预测的残差大小。一般来说预测差的 *MAD* 较小，用帧间编码所需比特数少。但若某些宏块预测得很不准(例如对处于物体运动后露出的背景区的宏块，它不是从上一帧中某位置移动过来的)，*MAD* 值会很大，预示着用帧间编码所需比特数不一定比帧内编码少，甚至会更大。这时可选择帧内编码模式。有时为了防止误码在帧间的扩散，也必须对画面中一定比例的宏块作帧内编码，以阻断因帧间预测造成的误码随时间的扩散。

(2) 由于采用了变长码，自适应量化步长调整，也由于图像中不同位置的子块有不同的复杂度和运动情况，编出的码流是不均匀的，有时编码器输出的比特数多，有时输出的比特数少。若希望它能在恒定速率的信道上正常传输，需设置码流缓存器。缓存器的充盈程度可以作为控制的依据。当缓存器接近上溢时，应把量化级调大，进行粗量化，从而使编出的码率降低，使缓存远离上溢状态。

(3) 做运动估计时，所用的参考帧可以是上一个输入帧，也可以是经本地解码后复原的、存放在帧存储器中的复原帧。而在做运动补偿时，参考帧必须是复原的、有量化误差的上一帧，以防止因帧间预测造成的漂移现象(见§6.4节)。

2. H.263，H.263＋，H.263＋＋

为满足在码率低于 64 kbps 的普通公用电话网或移动电话网上进行可视电话通信的需要，ITU-T 在对 H.261 进行改进的基础上提出了 H.263，它采用了半像素精度的运动估计，运动估计搜索范围也有所扩大，使压缩比有明显提高。其应用也可扩展到大于 64 kbps 的信道。

H.263 有 4 个选项：

(1) 无约束运动矢量(UMV)。

(2) 基于上下文的算术码(Syntax-based Arithmetic Coding)。

(3) 高级预测模式(APM)和重叠块运动补偿(OBMC)。它允许运动估计以8×8块为单位进行而不是只能以 16×16 宏块来进行。在宏块内像素运动一致性难以保证时，这样做可提高预测精度。而 OBMC 对减小方块效应有利。

(4) PB 帧模式。

H.263＋与 H.263 相比进一步扩大了应用范围和提高了压缩效率。它有 12 个选项，包括：增加了新的图像格式，运动矢量范围的进一步扩展，RVLC，可分级模式中的 B 帧、EI 帧和 EP 帧，先进帧内编码模式，去块效应滤波，改进的 PB 帧，参考图像选择，低分辨率刷新，另一种帧间 VLC，改进的量化模式，时域空域 SNR 可分级性等。这些技术不在此一一详细解释，总之，是可以提高编码的效率、灵活性和抗干扰性以适应多种应用的要求。而 H.263＋＋ 在H.263＋的基础上又增加

了四个选项,包括增强的参考图像选择模式、数据分割宏块组模式等,还定义了一些"类"(Profile)和一些附加增强信息。

3. H.264

H.264是由ITU-T第16专家组VCEG的H.26L发展而来,2001年VCEG和ISO/IEC JTC1/SC29/WG11组成了联合视频工作组(JVT),共同制订H.264,并于2003年完成。它也作为MPEG-4的第十个部分。是目前性能最好但也是最复杂的视频编码标准。它有很高的编码效率,在相同的复原图像质量情况下,其比特率可以比H.263降低50%。它的抗误码能力强,适于在移动和无线因特网等错误多发环境下传输。其总体结构分为二层:视频编码和数据分割层以及网络抽象层(NAL),以适应在多种广播、电信网上的传输。H.264的编码方案仍然是混合编码框架,但有很多改进和不同:

(1) 改进了运动补偿帧间预测的性能。采取的措施包括在必要时允许以更小的块为单位进行块匹配运动估计和补偿,最小可以4×4块为单位来进行。这使得在复杂运动区的运动补偿更准确。提高运动矢量精度,达1/4和1/8像素;B帧加权预测;允许采用多个参考帧等。采用多参考帧不仅可提高预测精度,也有利于降低误码扩散。

(2) 采用帧内预测技术,显著提高帧内编码块的编码效率。

(3) 采用4×4整数变换、性能近于DCT,但可用16比特整数运算来实现。

(4) 改进的环路内去块效应滤波器。

(5) 在熵编码方面,对4×4整数变换系数用CAVLC,对其他信息用UVLC编码。也可以用上下文自适应二进制算术码(CABAC)来进行编码。

H.264的其他特征如:FMO、SP和SI宏块组等等。

H.264有三个类(Profile):基本类(Baseline)、主类(Main)和扩展类(Extended)。容易引起混淆的是这三个类并不完全是全集和子集关系。例如有些包含在基本类中的编码工具并未包含在主类中(如一些差错控制措施等)。

H.264内容十分丰富,限于篇幅不可能在此逐一加以说明。有兴趣的读者可参阅H.264文本或有关文献。

6.6.2 MPEG系列标准

ISO/IEC的活动图像专家组(MPEG)制订了一系列MPEG标准,它们在数字电视领域具有深远的影响。

1. MPEG-1

MPEG-1是为数字存储媒体(DSM)的存储与检索而制订的活动图像、声音

及其组合编码的标准。码率为1.5 Mbps,图像质量和家用录像带(VHS)相当,广泛应用于VCD中。它是一个系统级标准,不仅有视频,也有音频和系统层的协议,包括复用、同步等,考虑得较完备,MPEG-2和MPEG-4也是系统级的标准。这和ITU-T制订的H.26X系列只含视频编码部分不同。我们在本节只介绍其视频编码部分的内容。在MPEG-1中采用了双向运动补偿技术,在前向预测帧(P帧)的基础上,增设了I帧(帧内编码帧)和B帧(双向预测帧)。对I帧只作帧内编码,不作帧间预测。它可作为后续P帧和B帧的参考帧。它提供了对码流进行随即访问的切入点。对P帧只作前向帧间预测编码,其运动补偿精度一般为0.5像素(亚像元精度)。它可作为后续P帧和B帧的参考图像。B帧是MPEG-1的创新,由于它可进行双向预测帧间运动补偿编码,提高了预测精度,从而使编码效率得到提高。为了解决后向预测的时间上非因果性问题,必须在编码前对帧的次序进行重排。

2. MPEG-2

MPEG-2在MPEG-1的基础上又有了进一步的发展。它是一个通用的标准,适于许多视频源的编码。对不同分辨率的逐行和隔行扫描的图像信源都可适用。它通过"类"(profile)和"等级"(Level)为基础的规范化,可满足不同的图像分辨率及有关的存储成本和处理速度的需要。它广泛应用于数字电视各应用领域,如:DVD,通过卫星、有线、地面传输的标准清晰度(SDTV)和高清晰度(HDTV)数字电视的广播,点播电视(VOD),数字声音广播(DAB)等。在1994年标准正式公布后还通过增加新的Part的方式继续补充新的内容,如:DSM-CC,AAC等。

3. MPEG-4

MPEG-4和MPEG-1和MPEG-2的本质区别是MPEG-4是面向对象的,而不仅仅是面向整幅图像的,因而更加灵活。它提供了基于内容的AV信息压缩存贮处理与操作,向各种基于网络的AV业务提供技术支持。它将内容与交互作为核心,为多媒体数据压缩提供了更为广阔的平台。它是一个工具包,更多定义的是一种格式和框架,从而可以容纳各种新技术和算法来适应新的应用。MPEG-4并不针对某项特殊的应用,而是尽量支持对多种应用均有帮助的功能组。MPEG-4支持的功能可分为基于内容的交互性,高压缩率和通用存取三大类。它的视频压缩技术主流仍是分块DCT,在对象边界处则为形状自适应DCT,即SA-DCT,运动补偿帧间预测,算术码等。对静止图,可采用小波变换编码。其Sprite编码主要针对静止背景。二维和三维物体可以用网格编码并利用SNHC编码将自然和人工合成的对象进行混合编码,为多媒体节目制作提供了有力工具。

为了从序列图像中提取出有意义的对象,需要进行图像分割。MPEG-4的内

容并不包括图像分割的具体方法。由于用计算机自动分割难以获得完美的分割结果,这成为基于对象编码推广应用中的瓶颈和制约因素。MPEG-4 的编码效率和 H.263 大致相当。

6.6.3 JPEG、JPEG2000 和 JPEG-LS

1. JPEG

JPEG 是由 ISO 和 CCITT 联合制订的,标准的制订由联合工作组 ISO/IEC JTC1/SC2/WG8 负责。它的基本系统是基于分块 DCT 变换编码的有损压缩算法。其扩充系统含分层编码、渐进模式和基于空间预测的无失真工作模式。目前 JPEG 被广泛用于数码相机、网上图像传输、Telematic 业务图像通信终端。它也被用于以帧内编码方式传输活动图像,即 MJPEG(Motion JPEG),MJPEG 在数字电视节目编辑、制作等方面也得到了广泛应用。

2. JPEG2000

JPEG2000 是于 2000 年制订的,由于采用了小波变换、EBCOT 等新技术,其压缩比要比基于 DCT 的 JPEG 有明显提高。JPEG2000 的优越的抗误码性能,可实现渐进传输,和感兴趣区等特点大大拓展了它的应用范围。

3. JPEG-LS

它是对连续调静止图像进行无损或近于无损压缩编码的国际标准。适用于医学、遥感等应用领域的图像传输与存储。

虽然在 JPEG 标准中也规定了无损压缩方法,但编码效率不高。JPEG-LS 比它有更高的编码效率。JPEG-LS 采用的主要技术有自适应帧内预测、行程码、Colomb 编码。在进行近于无损编码时还包括对预测差的量化(较精细的量化)。

此外,JBIG 是二值图像编码标准,它可用于 4G 传真机等应用领域。其技术基础是自适应算术码。对 JBIG,本节不再作进一步的说明。

6.7 小波变换编码

6.7.1 嵌入式零树小波算法 EZW(Embeded Zerotree Wavelet)

这是由 J. M. Shapiro 在 1993 年提出的,被认为是一种比较好的利用小波变换的压缩方法。后来又有 A. Said 等人提出了零树算法的改进算法,使算法的效

率进一步提高。

这一算法的基本特征是：

(1) 将小波变换系数按大小排序，按系数的重要性依次传输，即嵌入式算法的基本思想；

(2) 利用图像在不同尺度下小波变换子图像之间自相似性(Self Similarity)，构造小波系数的零树结构。

1. EZW 算法的基本思想

嵌入式零树小波编码算法(EZW)是一种简单高效的嵌入式编码算法。嵌入式编码的最大特点是编码器可以在任何时候停止编码，而解码器仍能正确解码，图像传输是渐进的，从无到有，随着解码过程的进行，显示的图像越来越清晰。在图像浏览等实际应用中，渐进式的图像传输十分有用，而且 EZW 算法计算简单，充分利用了小波系数零树结构。

EZW 算法的基本思想是优先传送重要的系数，即幅值大的系数。当传送一个系数时，从它的最高位开始传送。先给定一个较大的门限，找出相对于该门限重要的系数，然后传送这些系数的最高位；随后不断降低门限，对于每一门限，都从不重要的系数中找出新的重要系数，并对已经找到的重要系数不断细化，从而完成图像的渐进传输。这里产生了一个问题，每次传送重要系数时，同样需要传送它的位置信息，这将大大增加编码开销，EZW 利用零树结构，并依靠巧妙地安排扫描系数的顺序，大大降低了编码重要系数位置信息的开销。

图 6.15 示出一个三层小波分解的情况，共产生 10 个不同分辨率的子图。每个子图是它下一级(高一级分辨率)子图的四分之一大小，即一个像素对应下一级子图中 2×2 个像素，而且同一方向上的子图有较强的“自相似性”。高频子图中的小波系数反映的是图像中的高频分量，即灰度有突变的奇异点、边界、纹理等。因此，高频子图中大部分小波系数在 0 值附近。

设小波系数大小为 $x(i,j)$，设阈值为 T，若 $|x(i,j)|\geqslant T$，则称 $x(i,j)$ 为重要系数(非零系数)。反之，为不重要系数。

零树算法的基本假设是：如果在粗分辨率下某个小波系数 $x(i,j)$ 对某个阈值 T 是不重要的，那么在高分辨率下，同样空间方向，同样空间位置上的所有小波系数很可能都是不重要的。

在图 6.15 中，除了最高频子带(最高分辨率)以外，某一尺度高频子图的每一个系数均相当于下一级子图中 4 个系数(在同一方向、同一空间位置上)。较粗分辨率下的系数为“父”，在较细分辨率下的 4 个系数为“子”。这样形成的一棵树，称为小波树。注意：最低分辨率的低频子图 LL_J 的每个系数只有 3 个“儿子”。

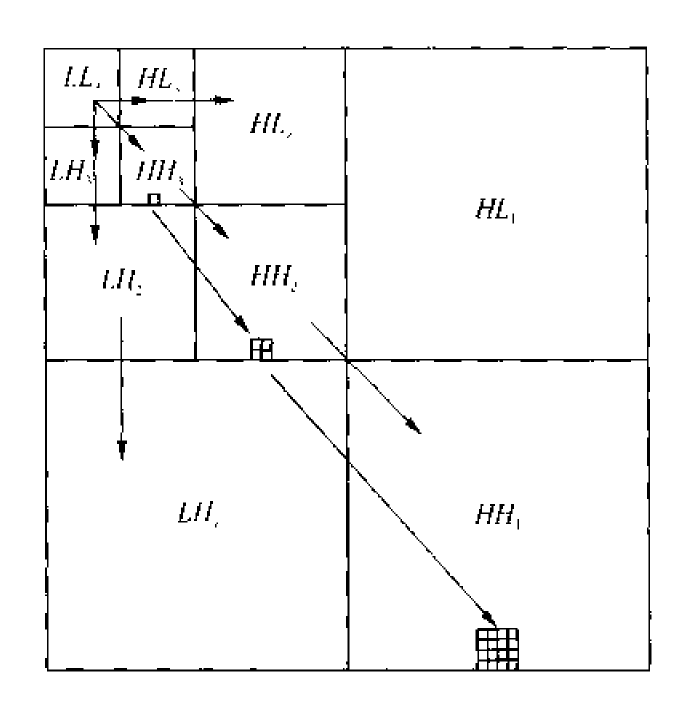


图 6.15 小波树示意图

设一阈值 T 来判决一个系数是否是重要的(或称有意义的)。如果一个系数和它所有后代对于 T 来说都是不重要的,那么这个系数就是零树的一个元素。如果对于 T,零树中的一个元素不是较粗一级中已发现的零树元素的后代,这个元素就是零树的根。如果一个系数是不重要的,但它又有一些重要的后代,这个系数称为孤立"零"。

这样,整个子图中的所有系数只需用 4 种符号来表示:

①零树根(ZTR);②孤立零(IZ);③正重要系数(POS);④负重要系数(NEG)。

2. EZW 算法的编码步骤

EZW 算法的大体过程是:先确定一组门限 $T_0,\cdots,T_{N-1}$,满足 $T_i=T_{i-1}/2$,且对于任意小波系数 x,有 $|x|<2T_0$。在编码过程中始终保持两张小波系数表,主表和从表。在任何时候,主表中那些尚未被确定为重要的系数(即目前还是不重要系数)在主表中的排列顺序与最初的扫描顺序相同。扫描是按一个子带一个子带的进行,在每个子带中扫描也按固定顺序从左到右、从上到下地进行。从表保存了目前已被确定为重要系数的小波系数的幅值。对于每个门限 T_i,主表和从表分别被扫描一次。

在主扫描过程中,将那些尚未被确定为重要的系数与 T_i 比较,判断它是否重要,并对判断结果进行编码输出,同时更新主表。如前所述,判断一个系数是否重要的流程如图 6.16 所示。

每次当一个系数被编码为重要系数,则将其附加到从表的最后,并将小波系数矩阵中该系数置为零,以便于在以后的主扫描中这些系数不会妨碍编码某一系数

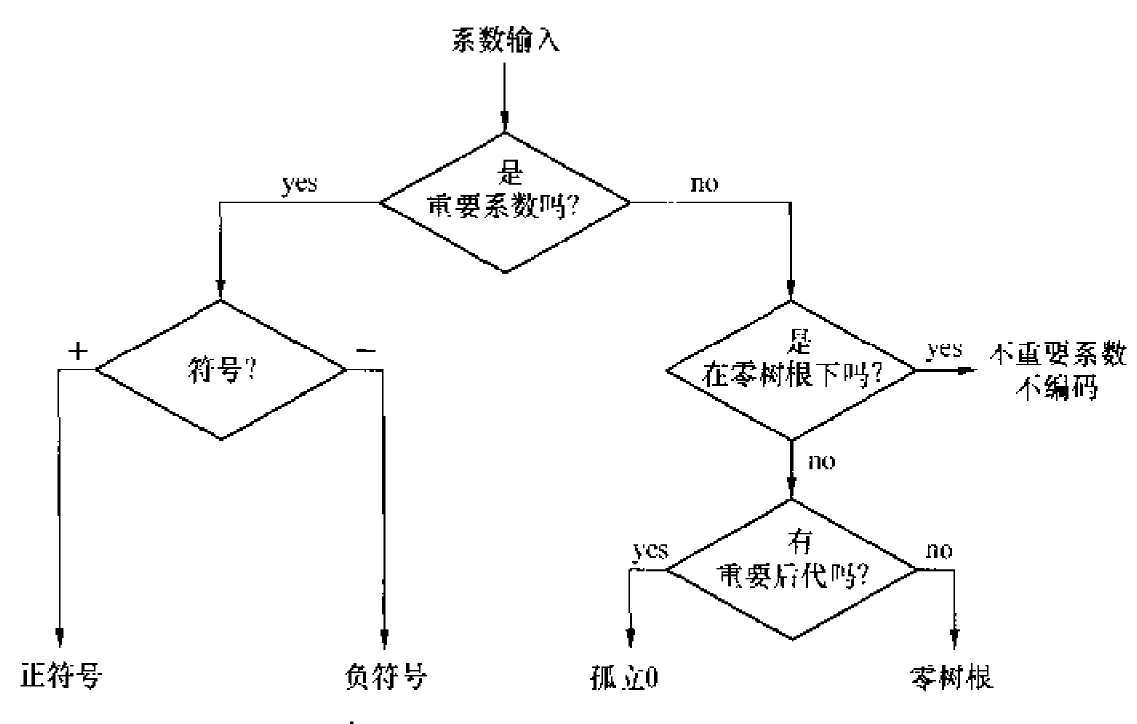


图 6.16　一个小波系数的编码

为零树根。

一次主扫描后接着进行一次从扫描，从扫描只从头到尾扫描从表中的系数，对每个系数输出符号“1”或符号“0”，使每个重要系数的不确定的范围缩小为原来的一半(即增加 1 比特的精度)。最初每个重要系数的不确定范围为 T_0。每进行一次从扫描，解码端就可以进一步细化已收到的系数的幅值，解码出的图像也越来越清晰。每进行一次从扫描后，编码器对从表中的系数进行降序排序，但排序过程要保证解码端也有能力进行同样的排序。

主扫描和从扫描反复进行下去，直到达到总码率的要求或整幅图像已无失真的全部编码完毕。

3. 数字实例

下面给出了该算法的一个简单例子，表 6.12和表 6.13 为对图 6.17 所示小波系数矩阵进行一次主扫描和从扫描后主表和从表的状态。为便于理解，对表 6.12 和 6.13 构建中的某些关键环节作如下说明：

(1) 取 $T_0 = 32$。系数 63 大于门限 32，所以产生符号 POS(positive)，当收到这个符号后，解码器可以确定该系数值位 F[32, 64)，并

63	34	49	10	7	13	−12	7
−31	23	14	−13	3	4	6	−1
15	14	3	−12	5	−7	3	9
−9	−7	−14	8	4	−2	3	2
−5	9	1	47	4	6	−2	2
3	0	3	2	3	2	0	4
2	−3	6	−4	3	6	3	6
5	11	5	6	0	3	−4	4

图 6.17　一个简单的 3 级小波分解系数矩阵的例子

得到重建值 48。将系数 63 放入从表，然后在系数矩阵中将它置 0。

(2) 系数 −31 小于门限，但它的后代中有一个值为 47 的系数，所以产生符号 IZ(Isolated Zero)。

(3) 系数 23 小于门限，且它的所有后代都小于门限，所以产生符号 ZTR(Zero Tree Root)。

(4) 在分辨率最高的子带中，可以将符号 IZ 和 ZTR 合并为一个符号 Z(Zero)。

(5) 系数 47 大于门限，产生符号 POS，然后这个位置的系数被置零，当进行下一次主扫描时，门限值为 16，则该系数的父系数 14 将被编码为 ZTR。

(6) 在进行完第一次主扫描后，从表中被加入了 4 项，随后进行对从表的扫描。对从表中的每一项所含系数根据其大小编码产生符号“1”或“0”，从而使每个系数的不确定区间从[32, 64)缩小为[48, 64)或[32, 48)。

(7) 在第一次从扫描结束后，对从表中的 4 项重新排序，顺序由(63, 34, 49, 47)变为(63, 49, 34, 47)，这里需要指出的是，重排的参考顺序是根据由从扫描输出的比特后的重建的幅值进行的，解码端在收到从扫描的输出后，解码器有能力进行同样的排序。

(8) 在第一次主、从扫描结束后，开始以 16 为门限进行第二次主、从扫描。在第二次主扫描中，LH3 中的 −31 产生 NEG，HH3 中的 23 产生 POS，HL2 中的(10, 14, −13)和 LH2, HH2 中的全部系数均被编码为 ZTR(or Z)。第二次从扫描后，从表中包含 6 项，依次为(63, 49, 34, 47, 31, 23)，它们的不确定区间分别为[48、64)，[32, 48)和[16, 32)。第二次从扫描将进一步细化这些区间，在扫描结束后，从表中各项重新排序为(63、49, 47, 34, 31, 23)，解码端可以得到重建值(60, 52, 44, 36, 28, 20)。

(9) 主、从扫描的过程将继续下去，编码器可以在任何时候停止编码。

表 6.12 主 表

子带	系数值	符号	重建值	子带	系数值	符号	重建值
LL3	63	POS	48	LH2	−9	ZTR	0
HL3	34	NEG	−48	LH2	−7	ZTR	0
LH3	−31	IZ	0	HL1	7	Z	0
HH3	23	ZTR	0	HL1	13	Z	0
HL2	49	POZ	48	HL1	3	Z	0
HL2	10	ZTR	0	HL1	4	Z	0
HL2	14	ZTR	0	LH1	−1	Z	0
HL2	−13	ZTR	0	LH1	47	POS	48
LH2	15	ZTR	0	LH1	3	Z	0
LH2	14	IZ	0	LH1	−2	Z	0

表 6.13 从 表

系数幅值	符 号	重建幅值	系数幅值	符 号	重建幅值
63	1	56	49	1	56
34	0	40	47	0	40

6.7.2 SPIHT 算法

A. Said 和 W. Pearlman 于 1996 年提出了 SPIHT 算法(Image Codec Based on Set Partitioning in Hierarchical Trees),改进了零树的思想。SPIHT 算法继承了 EZW 算法的特点,但在分割系数的方式和如何传输有效系数的位置信息上有所不同。它把对有效系数位置的传输隐含在算法的执行过程之中。同样码率下 SPIHT 算法的 PSNR 比 EZW 算法一般优 0.5—1dB。在 SPIHT 中小波系数按空间方向树组织,如图 6.18 所示。

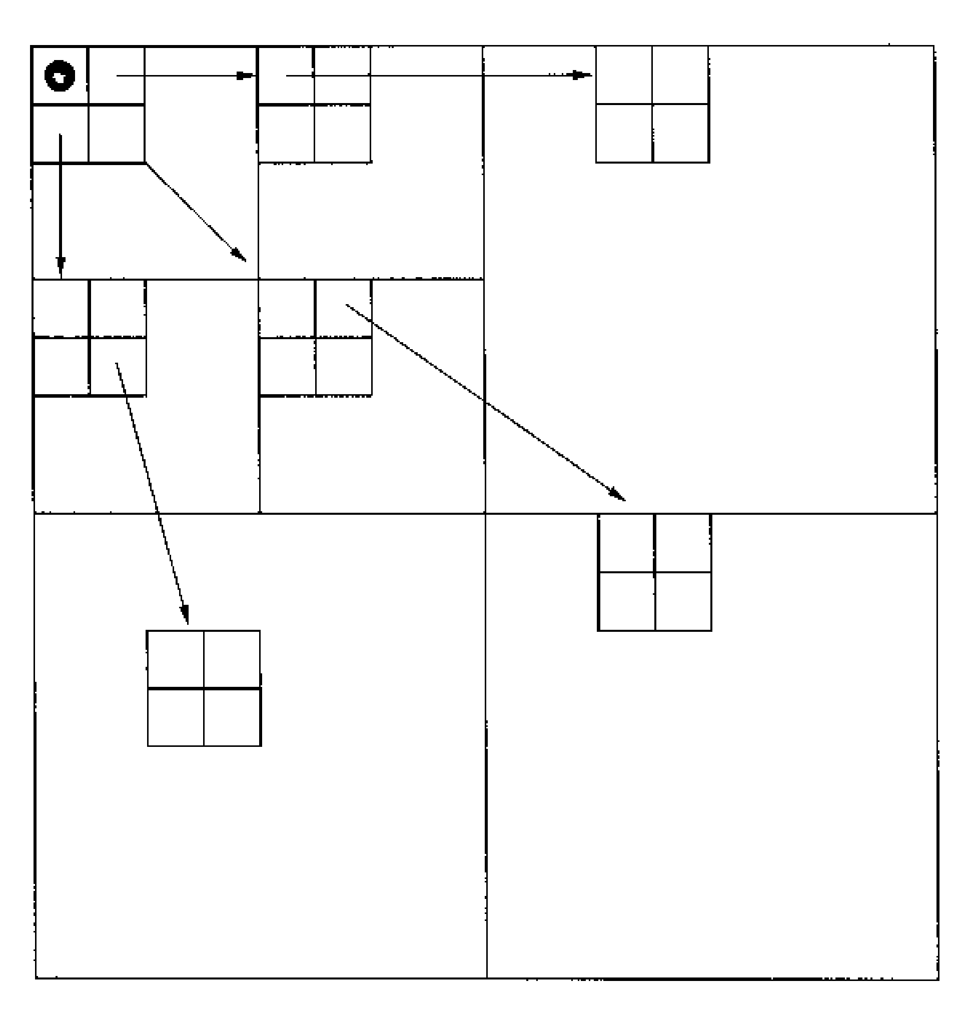


图 6.18 空间方向树结构示意图

空间方向树与 EZW 中讲的小波树基本相同,每个父节点有四个子节点。其不同在于树根,即 LL_J 在小波树中最高层的树根节点,每个只有三个子后代。而在空

间方向树中，将最高层 LL 系数分为 2×2 一组，其中三个有同一方向对应位置的四个子节点，而另一个没有后代。

在 SPIHT 算法中首先定义以下集合：

(1) H：空间方向树所有根结点的坐标集合；

(2) $D(i, j)$：点(i, j)的所有后代(子孙)的坐标集合，不包括(i, j)点本身；

(3) $O(i, j)$：点(i, j)的直接后代(儿子)的坐标集合；

(4) $L(i, j)$：点(i, j)除直接后代外所有后代坐标的集合；

$$L(i, j) = D(i, j) - O(i, j)$$

再定义一个判别函数 $S_n(\cdot)$，它的定义为：

$$S_n(\Gamma) = \begin{cases} 1, & \max\limits_{(i,j)\in\Gamma}\{|C_{i,j}|\} \geqslant 2^n \\ 0, & \text{其他} \end{cases} \tag{6.7-1}$$

式中：Γ 为集合，当为单一节点时，用其坐标(i, j)表示。

编码中使用了三个表：不重要系数表 LIP(List of Insignificant Pixels)、重要系数表 LSP(List of Significant Pixels)和不重要集合表 LIS(List of Insiginificant Sets)。LSP 初始化为空表，LIP 用最低频子带系数初始化，并将其中有后代的节点放入 LIS，作为初始化。

对重要图的确定主要是通过空间方向树的多次分裂来实现的。一个空间方向树 $T(i, j)$在初始化时分裂成树头结点 $c(i, j)$和剩余集合 $D(i, j)$。对 $c(i, j)$判断其重要性，若重要则转到 LSP 中。对集合 $D(i, j)$进行重要性测试，若$D(i, j)$是不重要的，则 $D(i, j)$用一个符号就可以表示出来。若 $D(i, j)$是重要的，则 $D(i, j)$继续分裂为两个集合 $O(i, j)$和 $L(i, j)$。对 $O(i, j)$中的每个元素分别进行重要性测试，把重要元素加入到 LSP 中，不重要元素加入到 LIP 中。对 $L(i, j)$集合进行重要性测试，若 $L(i, j)$不重要，则用一个符号就可以表示该集合，若 $L(i, j)$重要，则 $L(i, j)$分裂为四部分，每部分由相应空间方向树的位置上的元素构成，每一部分与 $O(i, j)$中的四个元素分别构成四棵新树，由于每棵新树的头结点已经判断，只对新树的剩余部分也就是 $L(i, j)$分裂出的四个集合进行判断。如此重复对每棵树进行分裂和判断直到找出每棵树中的所有重要元素，把它们转到 LSP 中。可以看到 SPIHT 算法对重要图的排序是通过一系列的集合分裂完成的，即一棵树 $T(i, j)$分裂成头结点元素 $c(i, j)$和剩余部分 $D(i, j)$，对重要的$D(i, j)$继续分裂成头结点的直接四个孩子 $O(i, j)$和剩余部分 $L(i, j)$，对重要的集合 $L(i, j)$再继续分裂为四棵新树的剩余部分。

对每棵树的分裂不是一次进行到底的，而是要按照一定的扫描顺序进行。对各个子带的扫描顺序与 EZW 算法的扫描顺序相同。对由最低频子带和头结点构

成的 LIP 中的元素是按从上到下、从左到右的顺序进行扫描的。而对其他子带则是按 2×2 的块为单位从上到下、从左到右依次扫描。对每个 2×2 块中元素还是按从上到下、从左到右顺序扫描。

SPIHT 编码算法流程如下：

(1) 初始化：

输出 $n=\lfloor\log_2(\max_{(i,j)}\{|C_{i,j}|\})\rfloor$，这里$\lfloor\cdot\rfloor$符号表示取小于·的最大整数。将 LSP 设置为空，将所有最高层系数(i, j)放入 LIP 集中，并将其中有后代的节点放入 LIS 中，并标以 A 类。

(2) 分类过程：

1) 对所有 LIP 中节点：

① 输出 $S_n(i, j)$。

② 如果 $S_n(i, j)=1$；则将(i, j)节点移至 LSP 中并输出 $C_{i,j}$的符号。

2) 对于所有 LIS 中的节点：

① 如果类型为 A 类，则：输出$S_n(D(i, j))$；

如果 $S_n(D(i, j))=1$，则：对所有 $O(i, j)$中节点(其坐标由 $C_{k,l}$表示)：

- 输出 $S_n(k, l)$；
- 如果 $S_n(k, l)=1$，则将(k, l)节点加入到 LSP 中并且输出 $C_{k,l}$的符号；
- 如果 $S_n(k, l)=0$，则将(k, l)节点加入到 LIP 中。

如果 $L(i, j)$非空，则将(i, j)节点移至 LIS 的尾部，并以 B 类标之，转向 2.2.2；否则，将(i, j)节点移出 LIS 集。

② 如果节点为 B 类，则：输出$S_n(L(i, j))$；

如果 $S_n(L(i, j))=1$，则：

- 将(i, j)节点的四个儿子节点以 A 类型加入到 LIS 集中；
- 将(i, j)节点移出 LIS 集。

(3) 细化过程：

对所有 LSP 中节点(i, j)，除去那些在最后一次分类过程中被加入的节点，输出$|C_{i,j}|$的第 n 个重要比特。

(4) 将 n 减 1，转向第 2 步。

解码算法和编码算法的结构是一样的，因而 SPIHT 不必显式地传送有效系数的位置信息，因为它已被隐含在算法的处理过程中。

习 题

1. 一个信源有 6 个字符，它们的概率如下：

$P(A)=\frac{1}{3}$, $P(B)=\frac{1}{4}$, $P(C)=\frac{1}{8}$, $P(D)=\frac{1}{8}$, $P(E)=\frac{1}{12}$, $P(F)=\frac{1}{12}$。

(1) 求这个信源的熵。

(2) 求一个平均长度最小的对字符 A, B, C, D, E, F 的二进制编码。这个编码的平均长度是多少? 冗余度多少?

2. 在一张 16×16 的离散图像上,上面一半(共 8 行)是白,下面一半是黑,现在把这张图用一维序列来表示(用两种方法):

① 逐行(由左向右)扫描,前行的尾和后行的首相接。

② 逐列(由上到下)扫描,逐列首尾相接。

把这样得到的一维序列记作 x_1, x_2, …, x_{n-1}, x_n…。

设这是一个各态历经的随机过程:

(1) 求它的一阶熵 $H(x_n)$。

(2) 分①、②两种情况求二阶熵 $H(x_n, x_{n-1})$,假定 x_0 为白色。

(3) 求条件熵 $H(x_n|x_{n-1})$。

(4) 假定白像素的灰度为 1,黑像素的灰度为 0,求图像的平均值和方差以及自协方差 $\sigma^2_{x_n x_{n-1}}$(分①、②两种情况讨论),并求两种情况下方差之比值。

(5) 如果取 $e_n=x_n-x_{n-1}$,求 $e(n)$的平均值和方差(分两种情况讨论),并求两种情况下方差之比值。

3. 一个信源,有 3 个字符,其概率为 $P(A)=0.8$, $P(B)=0.15$, $P(C)=0.05$,若对其进行霍夫曼编码和进行算术码编码,问上述哪种编码方法能取得更高的压缩比? 请从概念上说明你的回答的理由。

4. 试从概念上说明在做运动补偿帧间预测编码时,为什么运动补偿的参考帧必须是由本地解码器恢复的上一帧图像 f'_{k-1},而不能是上一个输入帧 f_{k-1}。

5. 试列举你所了解的图像编码国际标准,并说明它们各自采用了哪些图像压缩编码技术。

参 考 文 献

[1] Yun Q. Shi, Huifang Sun. Image and Video Compression for multimedia engineering, Fundamentals, Algorithms and standards[M]. CRC press, 2000.

[2] Ed. H.R. Wu, K.R. Rao, Digital video image quality and perceptual coding[M]. CRC Tayler & Francis, 2006.

[3] David Salomon.数据压缩原理与应用(第二版)[M].吴乐南,等译,北京:电子工业出版社,2003.

[4] 吴乐南.数据压缩[M].南京:东南大学出版社,2000.
[5] 沈兰荪,等.视频编码与低速率传输[M].北京:电子工业出版社,2001.
[6] 余松煜,等.现代图像信息压缩编码技术[M].北京:科学出版社,1998.
[7] 张旭东,等.图像编码基础和小波压缩技术——原理,算法和标准[M].北京:清华大学出版社,2004.
[8] 姚庆栋,等.图像编码基础(第二版)[M].杭州:浙江大学出版社,1992.
[9] 钟玉琢,等.MPEG-2运动图像压缩编码国际标准及MPEG的新进展[M].北京:清华大学出版社,2002.
[10] 钟玉琢,等.基于对象的多媒体数据压缩国际标准MPEG-4及其校验模型[M].北京:科学出版社,2000.
[11] William B. Pennebaker,等.JPEG静止图像数据压缩标准[M].黎洪松,等译.北京:学苑出版社,1996.
[12] 张春田,等.数字图像压缩编码[M].北京:清华大学出版社,2006.
[13] 毕厚杰.新一代视频压缩编码标准——H.264/AVC[M].北京:人民邮电出版社,2005.
[14] 小野定康,等.JPEG2000技术[M].北京:科学出版社,2004.

第7章　图像分割

前面几章所讨论的内容，如图像增强和图像复原，是对输入图像进行整体处理，处理后得到的仍是一幅图像，它是输入图像的改进。这种处理是为了便于人眼观察。但在对图像的分析研究和应用中，人们通常对图像中的某些特定目标（Object，在有些场合也称为“对象”）感兴趣，例如：汽车的车牌或者血液中的细胞等等。为了分析和识别这些目标，它们首先必须从图像中被提取出来。这种获得目标的技术被称为图像分割技术（Image Segmentation）。图像分割就是将图像分成若干具有特定意义的区域并将它们提取出来的图像处理技术。它是一种很重要的图像处理技术，是识别和理解等后续过程的必要前提，是从图像处理进到图像分析的关键步骤。例如，在交通监控系统中的车牌识别过程就是首先确定车牌的位置并分割出其中的字符和数字区域，然后对这些分割得到的区域进行识别。

图像分割技术近年来得到了很大的发展和深入的研究，也在实际中得到了广泛的应用，促进了一些新的工程技术的突破。这些应用中特别值得一提的是基于内容（Content）的视频编码协议 MPEG－4[1] 等和多媒体内容描述接口协议 MPEG－7[2]。基于内容的压缩编码不但在压缩效率上有很大的提高，而且图像质量也有较大的改善。在这些系统中，图像分割是其中关键一环。鉴于图像分割技术的重要性，本章将介绍一些经典的图像分割算法。7.1 节给出了本章大致的内容安排。

7.1　图像分割的定义和分类

图像分割的定义有不同的解释和描述，下面分别从定性的角度和集合的角度对图像分割进行定义。从定性的角度而言，图像分割就是把图像分成若干有特定意义的区域的处理技术，即依据图像的某些特性（如灰度、纹理等）把图像分成若干性质不同的区域，在每个区域内部有相同或者相近的特性，而相邻区域的特性不相同。这里，所谓“有意义”是一种很笼统的说法，就主观意愿来说，希望这些区域和场景中的各目标相对应，但这是十分困难的。分割过程中一般假设在同一区域内

特性的变化平缓，而在区域的边界上特性的变化剧烈。从上面可以看出，图像分割从本质上说是根据一定的准则将各像素进行分类的过程。这样可以借助于集合的概念对图像分割进行定义。

设一幅数字图像中所有像素的集合为R，P为逻辑谓词（分类准则），则图像分割将R分成N个子集$\{S_1, S_2, \cdots, S_N\}$，这些子集满足：

(1) $\bigcup_{i=1}^{N} S_i = R$；

(2) $S_i \cap S_j = \varnothing$，其中$i \neq j$；

(3) $P(S_i) = \text{TRUE}$；

(4) 对于$i \neq j$，$P(S_i \cup S_j) = \text{FALSE}$；

(5) 相对于S_i的那个区域是连通的。

其中，条件(2)表示分割结果中的每个子区域是互不重叠的。条件(3)表示同一个子区域内部具有相似的特性。条件(4)表示不同的子区域具有不同的特性。

现有的关于图像分割的算法非常多，对这些算法进行分类的方法也有不少。例如：根据待分割图像颜色属性，图像分割可被分为灰度图像分割和彩色图像分割；根据分割目标的变化状态，图像分割可分为静态图像分割和动态图像分割；根据分割目标的应用领域可分为工业图像分割、医学图像分割、军事图像分割、交通图像分割等；根据分割过程中算法实现策略的不同，分割算法可分为并行算法和串行算法；根据像素特性在目标区域内部性质一致性和区域间边界上不连续性，分割算法可分为基于边缘的分割算法和基于区域的分割算法。本章的内容安排主要采用了基于边缘和基于区域的这种分类方法对一些经典算法分别归类并介绍。

本章首先介绍基于边缘（7.2节）和基于区域（7.3节）这两类分割算法。为了提高分割性能，这两类分割算法有融合的趋势，在7.4节介绍了边缘和区域联合分割算法。图像分割评价方法作为图像分割的重要部分在7.5节介绍。由于篇幅的限制，本章略去了一些相当复杂的分割算法，如利用神经网络、模糊理论或遗传算法等理论的分割算法[3][4][5]，读者可查阅相关文献以了解这些算法。

7.2　基于边缘的图像分割

图像边缘是图像的基本特征之一，它是图像局部特性不连续（或突变）的结果，例如灰度的突变或者纹理的突变等等。图像边缘蕴含了丰富的图像内在信息（如方向和形状），它一般位于目标与背景之间、目标与目标之间和区域与区域之间。基于边缘的图像分割方法通过检测不同区域间的边缘来解决分割问题。本节介绍一些基于边缘的图像分割算法。

7.2.1 微分算子

图像边缘是图像基本特征之一。边缘可以定义为图像局部特性的不连续性，例如，灰度的突变，颜色的突变，纹理结构的突变等等。边缘通常意味着一个区域的终结和另一个区域的开始。边缘信息对信息分析和人的视觉都是很重要的。边缘的检测常借助空域微分算子进行，通过微分模板与图像卷积完成。本节主要讨论常用的边缘检测的微分算子。

在一幅图像中，边缘有方向和幅度两个特性。一般认为沿边缘走向的灰度变化较为平缓，而垂直于边缘走向的灰度变化剧烈。图像边缘可为阶跃状、斜坡状和屋顶状。这三种边缘示例见图 7.1。其中，图 7.1(a)为实际阶跃状边缘，图 7.1(b)为屋顶状边缘，图 7.1(c)为斜坡状边缘。

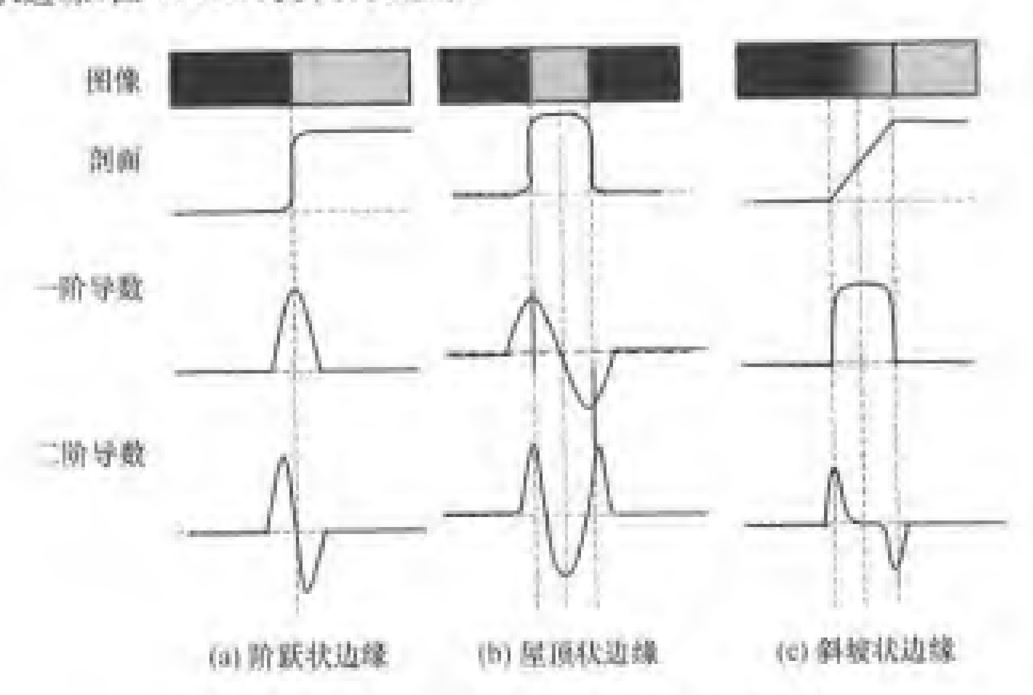


图 7.1 边缘灰度及其一阶、二阶导数变化示意图

由上可见，在边缘上灰度的一阶导数幅值较大，对于实际的边缘，其二阶导数在边缘上的值为零，但其左右分别为一正一负两个峰。因此，一阶导数可以用来判断图像点是否为边缘点，同时二阶导数的符号可以用来判断该边缘点是在边缘亮的一边还是暗的一边。在数字图像中用差分近似微分，按照一定方法算出各像素的一阶导数的大小(及方向)，然后通过取阈值等操作便可将边缘检出。阈值的选取决定了边缘检测的效果。上述边缘检测方法通常称为差分边缘检测方法。该方法是原始的经典的边缘检测方法。该方法要求差分的方向和边缘方向垂直，如果要进一步提高精确度，需要在不同方向上进行差分运算对比。

1. 梯度算子

对于数字图像,梯度(Gradient)是一阶导数的二维等效式。因此通过求取梯度可以判断边缘点。图像 $f(x, y)$在位置(x, y)的梯度向量定义为:

$$\nabla f = \begin{bmatrix} G_x \\ G_y \end{bmatrix} = \begin{bmatrix} \frac{\partial f}{\partial x} & \frac{\partial f}{\partial y} \end{bmatrix}^{\mathrm{T}} \tag{7.2-1}$$

式中:G_x 和 G_y 分别是梯度在 x, y 方向上的分量。我们知道梯度向量指向位置(x, y)处 $f(x, y)$的最大变化率方向。其幅度定义为:

$$\nabla f = |\nabla f| = \sqrt{G_x^2 + G_y^2} \tag{7.2-2}$$

方向定义为:

$$\alpha(x, y) = \arctan(G_y/G_x) \tag{7.2-3}$$

式中:α 角是梯度向量与 x 轴的夹角。

在实际应用中,有时为了避免平方和运算,也可用 x, y 方向上的两个分量的绝对值之和或最大值来近似梯度幅值,即

$$\nabla f \approx |G_x| + |G_y| \text{ 或 } \nabla f \approx \max\{|G_x|, |G_y|\} \tag{7.2-4}$$

对于数字图像,通常以两相邻点之间灰度差(一阶差分)来近似 G_x 和 G_y,即:

$$\Delta_x f = f(m, n) - f(m-1, n) \tag{7.2-5}$$

$$\Delta_y f = f(m, n) - f(m, n-1) \tag{7.2-6}$$

上述公式采用模板表示,则为:

$$\begin{bmatrix} 0 & 0 \\ -1 & 1 \end{bmatrix} \text{ 和 } \begin{bmatrix} 0 & -1 \\ 0 & 1 \end{bmatrix}$$

模板算子运算时采用类似卷积的方式,即把模板在图像上移动并在每点计算相应的梯度值。在边缘灰度值过渡较为尖锐并且噪声比较小时,梯度算子检测效果比较好。

除了上述的方法估算梯度值外,还有很多其他的算子模板估算梯度值。下面介绍另一种常用的梯度算子——罗伯特(Robert)梯度算子。在 7.2.2 曲面拟合边缘检测小节,我们还将介绍另外几种常用的梯度算子,这些梯度算子不同程度地提高了检测精度和抗干扰能力。

罗伯特梯度算子采用了对角方向相邻两像素之差来计算梯度值,即

$$\Delta_x f = f(m, n) - f(m-1, n-1) \tag{7.2-7}$$

$$\Delta_y f = f(m-1,\ n) - f(m,\ n-1) \qquad (7.2-8)$$

采用模板表示为：

$$\begin{bmatrix} -1 & 0 \\ 0 & 1 \end{bmatrix} \text{ 和 } \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix}$$

2. 拉普拉斯(Laplace)算子和LOG算子

边缘检测也可以用二阶微分算子来进行。拉普拉斯算子是一种二阶导数算子，它是一个标量而不是矢量。对一个连续函数 $f(x,\ y)$，它在位置 $(x,\ y)$ 处的拉普拉斯值(即二阶导数)定义为：

$$\nabla^2 f = \frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2} \qquad (7.2-9)$$

在数字图像中用差分来近似，它的一个表示式为：

$$\nabla^2 f = f(m+1,\ n) + f(m-1,\ n) + f(m,\ n+1) + f(m,\ n-1) - 4f(m,\ n) \qquad (7.2-10)$$

其模板为

$$\begin{bmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{bmatrix}$$

注意，模板所有系数之和为0。即，如果在模板内各相应位置处 $f(m,\ n)$ 的值相同(没有边缘)，则算子的响应为0。

在灰度阶跃边缘两侧，其值一边为正，一边为负；在斜坡形边缘处，其值为零，且在此零点处两侧也有一正一负两个峰。这一正一负的两个峰的大小和走向反映了边缘的强弱及走向。以往有人简单地取其绝对值，即取 $|\nabla^2 f|$，并和阈值相比较，把 $|\nabla^2 f| > t$ 的像素看成边缘点。这样所得到的边缘结果一般很不满意。仔细分析 $|\nabla^2 f|$ 可以看出，它对孤立噪声点的响应是对阶跃边缘响应的四倍，对单像素线条的响应是对阶跃边缘响应的两倍，对线端及斜向边缘的响应大于垂直及水平走向边缘的响应。所有这些缺点使拉普拉斯算子检测边缘的效果比较差。

从上述分析可知，拉普拉斯算子对噪声比较敏感，并且拉普拉斯算子不能检测边缘的方向，容易产生双边缘。为了克服上述不足，可先用一个高斯型二维低通滤波器对图像进行平滑，然后对图像求拉普拉斯算子。该方法的算子称 Laplace of Gauss 算子 $\nabla^2 G$ 或 LOG 算子。该算子是在拉普拉斯算子的基础上实现的，它得益

于人对视觉机理的研究。采用正态分布函数作为平滑函数。设高斯型滤波器的空间响应函数为：

$$g(x, y) = K'\exp\left(-\frac{x^2 + y^2}{2\sigma^2}\right) \tag{7.2-11}$$

式中：K'为常数；σ为标准差，它决定着对图像的平滑程度。

对图像 $f(x, y)$进行高斯滤波，得到：

$$f(x, y) * g(x, y) \tag{7.2-12}$$

然后再经拉普拉斯算子处理得到

$$\nabla^2 G = \nabla^2[f(x, y) * g(x, y)] = \nabla^2[g(x, y)] * f(x, y) \tag{7.2-13}$$

式中：
$$\nabla^2[g(x, y)] = K\left[\frac{x^2 + y^2}{\sigma^2} - 2\right]\exp\left(-\frac{x^2 + y^2}{2\sigma^2}\right)$$

式中：$K = K'/\sigma^2$。LOG算子实际是以$\nabla^2 g(x, y)$为卷积核，对原灰度图像进行卷积运算后提取的零交叉点为边缘点。在实现时，K应取大一些，使得$\nabla^2 G$运算可以用整系数卷积运算来实现。$\nabla^2 g(x, y)$是以(0, 0)为中心旋转对称的，函数形状像墨西哥草帽，故又被称为墨西哥草帽算子，其截面图如图7.2。

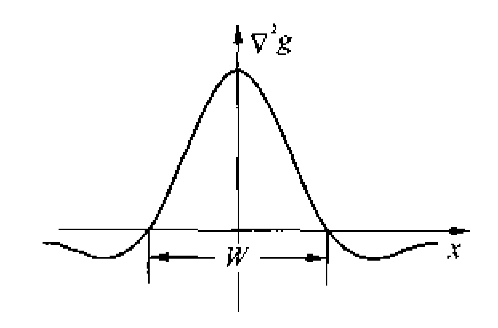


图7.2 墨西哥草帽算子的函数截面图

它的正瓣宽度 W 为：$W = 2\sqrt{2}\sigma$。因为$\nabla^2 g(x, y)$有无限拖尾，在具体实现 f 与$\nabla^2 g(x, y)$之间的卷积运算时应取一个 $N \times N$ 的空间窗，在窗内进行卷积。显然，N 不能太小，以免过多地截去其拖尾。N 与$\nabla^2 g(x, y)$的主瓣宽度 W 有关，例如 $W = 9$ 时，取 $N = 35$ 为宜，这时$\nabla^2 g(x, y)$函数形状反映为 $N \times N$ 模板内的各个系数。窗口内各个系数之和应为0。理论和实践证明，要想用这种方法得到较好的边缘检测，需要采用较大的窗口；为了减小卷积运算的计算量，可以用两个不同宽度的高斯曲面之差来近似$\nabla^2 g(x, y)$，即用算子DOG来近似代替$\nabla^2 g(x, y)$。关于DOG具体参阅文献[6]。DOG算子具有较好的性能，这是因为对人眼视觉机理的研究表明，视网膜神经节中有一类细胞，其输出相当于两个高斯卷积之差，该方法能较好地反映人的视觉特性。

对于数字图像，LOG算子有两种实现方法。一种是先做高斯滤波，再对滤波后的图像求拉普拉斯卷积。另一种是先求高斯滤波器的拉普拉斯变换，再与图像进行卷积。LOG算子可采用下面的5×5的模板，即

$$\begin{bmatrix} -2 & -4 & -4 & -4 & -2 \\ -4 & 0 & 8 & 0 & -4 \\ -4 & 8 & 24 & 8 & -4 \\ -4 & 0 & 8 & 0 & -4 \\ -2 & -4 & -4 & -4 & -2 \end{bmatrix}$$

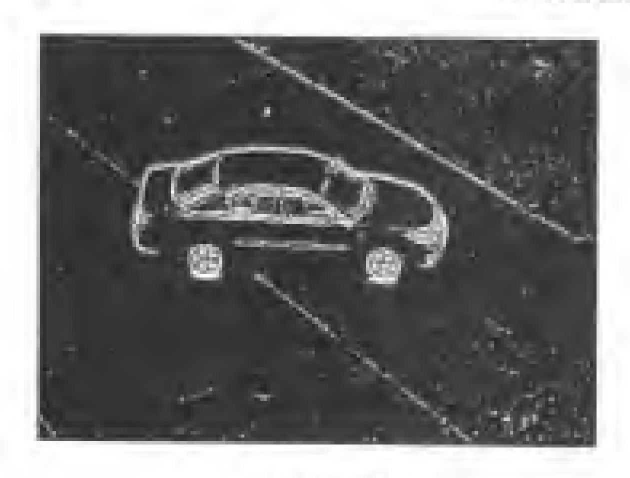
上述算法由Marr和Hildreth提出，故可称为M-H算法，也有人称之为边缘检测的二阶导数法。采用拉普拉斯算子和LOG算子分别对图7.3(a)所示的Car图像进行边缘检测。图7.3(b)和(c)分别是用Laplace模板和LOG模板与Car图像卷积得到的结果，图中黑的像素对应着零值，亮的像素对应着正值，灰色对应着负值。通过比较这两个结果，可以看出LOG算子的检测性能要优于拉普拉斯算子。



(a) Car原图



(b) 拉普拉斯算子检测



(c) LOG算子检测

图7.3 拉普拉斯算子和LOG算子性能比较

7.2.2 曲面拟合边缘检测

拟合是指用某个解析函数逼近实际的数据。曲面拟合的基本思想是用一个平滑的曲面与待检测点周围某邻域内像素的灰度值进行拟合，然后计算此曲面的导数(可以用一阶或二阶导数)，但通常用的是该点的梯度(与一阶导数的等效)，因此实际上根据此方法可以求得精度较高的待检测点的梯度，从而检测到边缘。该方法可减小噪声等影响，得到比 Robert 算子更好的边缘检测结果。

1. 平均值差分

先看在四个像素点上拟合一张平面的最简单情况，以便熟悉曲面拟合法的步骤。设在 xy 图像平面上有一个四点的区域，设为

$$a = \{(x, y), (x, y+1), (x+1, y), (x+1, y+1)\}$$

用 $f(x, y)$、$f(x, y+1)$、$f(x+1, y)$、$f(x+1, y+1)$ 表示这四个点的灰度值，现用一个平面 z 来拟合这四个点的灰度值，即用

$$z(x, y) = ax + by + c \tag{7.2-14}$$

来近似 f，使均方误差

$$\varepsilon^2 = \sum_{x, y \in a} [z(x, y) - f(x, y)]^2 \tag{7.2-15}$$

为最小。为此，将式(7.2-15)分别对 a、b、c 微分并使它们等于零，解得参数 a、b 和 c 的值。容易求得

$$a = \frac{1}{2}\{f(x+1, y) + f(x+1, y+1) - f(x, y) - f(x, y+1)\} \tag{7.2-16}$$

$$b = \frac{1}{2}\{f(x, y+1) + f(x+1, y+1) - f(x, y) - f(x+1, y)\} \tag{7.2-17}$$

对平面 z 求梯度，则 $\frac{\partial z}{\partial x} = a$，$\frac{\partial z}{\partial y} = b$，利用上述的求梯度幅值的公式，可以求得梯度幅值。这里 a 是两行的平均值差分，b 是两列的平均值差分，即先平滑(求平均)然后求差分，所以有一定的抑制噪声的作用。采用模板来表示式(7.2-16)和式(7.2-17)，则有

$$\begin{bmatrix} -1 & -1 \\ 1 & 1 \end{bmatrix} \quad 和 \quad \begin{bmatrix} -1 & 1 \\ -1 & 1 \end{bmatrix}$$

2. 普莱惠特(Prewitt)算子

把平面 $z(x, y)$ 改为二次曲面,即

$$z(x, y) = ax^2 + bxy + cy^2 + dx + ey + g \tag{7.2-18}$$

并把进行拟合的区域加大,在此,我们增加为 3×3。采用和上小节完全类同的步骤,最后求得梯度幅值。采用 3×3 模板表示为:

$$M_x = \frac{1}{3}\begin{bmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{bmatrix} \quad 和 \quad M_y = \frac{1}{3}\begin{bmatrix} 1 & 1 & 1 \\ 0 & 0 & 0 \\ -1 & -1 & -1 \end{bmatrix}$$

3. Sobel 算子和 Kirsch 算子

上面的算子是平均后求微分。与上述推导相似,Sobel 算子是先作加权平均然后再微分,它的水平和垂直分量的模板分别为:

$$\begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix} \quad 和 \quad \begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix}$$

Sobel 算子的检测效果较好,具有一定的抗噪能力,在检测阶跃边缘时得到的边缘宽度至少为两个像素。

Kirsch 算子是一个 3×3 的非线性算子。其基本思想是希望改进取平均值的过程,尽量使边缘两侧的像素各自与自己同类的像素取平均值,然后求平均值之差,从而减小由于取平均值而造成边缘细节丢失。设存在点 (m, n),该点的邻域如图 7.4,则像素经 Kirsch 算子处理后变为:

A_0	A_1	A_2
A_7	$f(m, n)$	A_3
A_6	A_5	A_4

图 7.4 Kirsch 算子的邻域

$$g(m, n) = \max\left\{1, \max_{i=0}^{7}\left[\,|5S_i - 3T_i|\,\right]\right\} \tag{7.2-19}$$

式中:

$$S_i = A_i + A_{(i+1)8} + A_{(i+2)8}$$

$$T_i = A_{(i+3)8} + A_{(i+4)8} + A_{(i+5)8} + A_{(i+6)8} + A_{(i+7)8}$$

式中:i 从 0 到 7,这里 A 的下标以模 8 进行计算,这个算子只考虑 (m, n) 点 8 邻域的情况,与 $f(m, n)$ 无关。如果在 3×3 邻域内有边缘存在,则边界线将邻域分成

三个点的 S_i 集及五个点的 T_i 集，并使它们处于不同的区域。通过改变 i、$i=0$, 1, 2, …, 7，这样寻找最大值，这种组合方式意味着边缘最大可能的走向。如有边缘存在，$|5S_i-3T_i|$ 较大(大于 c 值)，这时 $g(m, n)$ 中就包含有存在边缘及其强度的信息；若邻域内没有边缘，则 $|5S_i-3T_i|$ 的值较小(小于 c)，$g(m, n)$ 就取预定的较低的固定背景值 c。通常采用八方向 Kirsch 模板的方法进行检测，取其中最大的值作为边缘强度，而将与之对应的方向作为边缘方向。常用的八方向 Kirsch 模板如下，各方向间的夹角为 45°。

$$\begin{bmatrix} -5 & 3 & 3 \\ -5 & 0 & 3 \\ -5 & 3 & 3 \end{bmatrix} \begin{bmatrix} 3 & 3 & 3 \\ -5 & 0 & 3 \\ -5 & -5 & 3 \end{bmatrix} \begin{bmatrix} 3 & 3 & 3 \\ 3 & 0 & 3 \\ -5 & -5 & -5 \end{bmatrix} \begin{bmatrix} 3 & 3 & 3 \\ 3 & 0 & -5 \\ 3 & -5 & -5 \end{bmatrix}$$

$$\begin{bmatrix} 3 & 3 & -5 \\ 3 & 0 & -5 \\ 3 & 3 & -5 \end{bmatrix} \begin{bmatrix} 3 & -5 & -5 \\ 3 & 0 & -5 \\ 3 & 3 & 3 \end{bmatrix} \begin{bmatrix} -5 & -5 & -5 \\ 3 & 0 & 3 \\ 3 & 3 & 3 \end{bmatrix} \begin{bmatrix} -5 & -5 & 3 \\ -5 & 0 & 3 \\ 3 & 3 & 3 \end{bmatrix}$$

4. Wallis 检测法

这种非线性边缘检测算子是建立在同态图像处理技术的基础上的。如果一个像素其亮度的对数值与其四邻亮度对数值的均值之差超过某个门限，则表示在该像素处存在边缘。参照图 7.4 的符号。此算子定义为

$$\begin{aligned} g(m, n) &= \left| \log(f(m, n)) - \frac{1}{4}[\log A_1 + \log A_3 + \log A_5 + \log A_7] \right| \\ &= \left| \frac{1}{4}\log \frac{[f(m, n)]^4}{A_1 A_3 A_5 A_7} \right| \end{aligned} \tag{7.2-20}$$

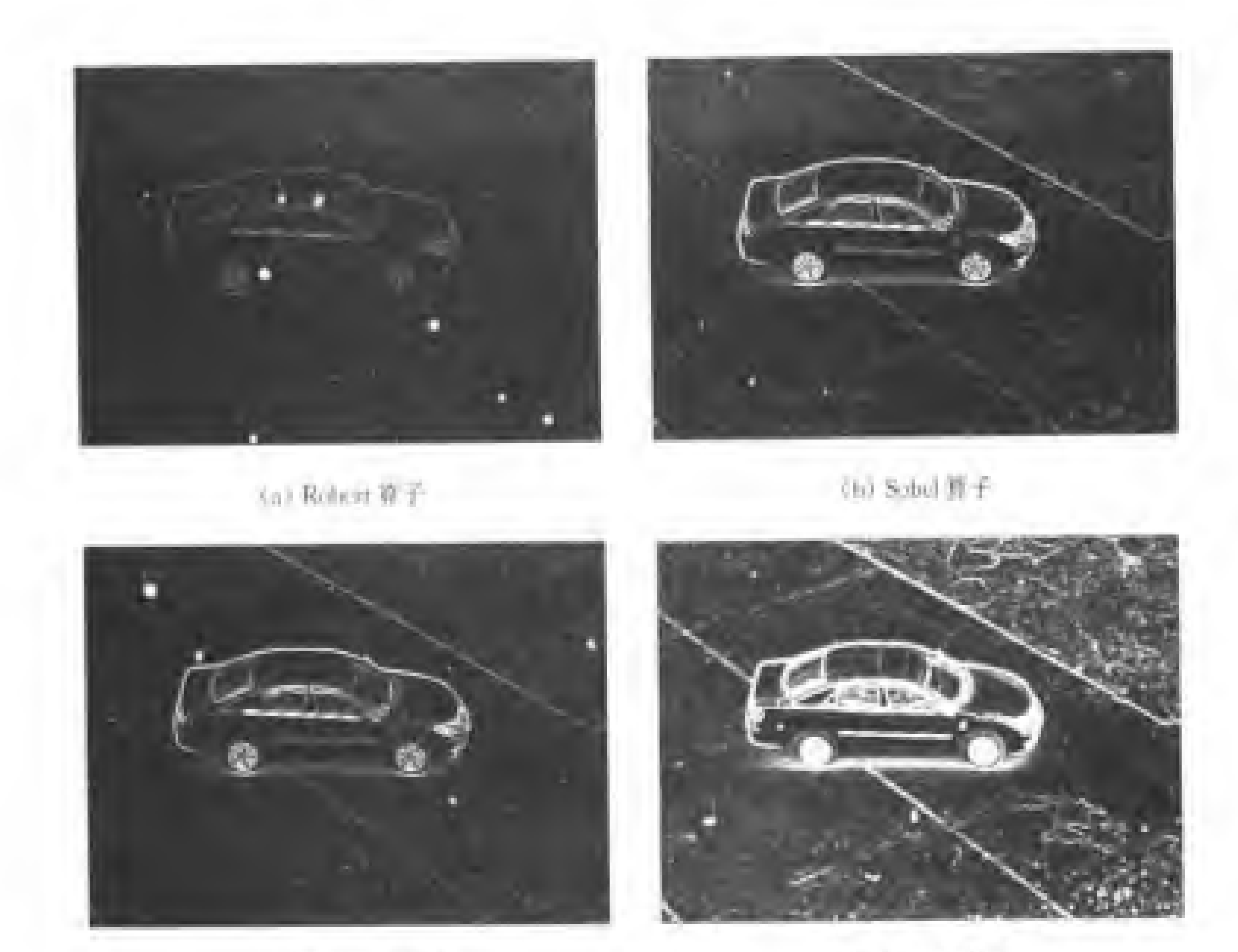
该算子的主要优点在于它对亮度的倍增变化不敏感，即对背景亮度的缓慢变化不敏感。另外，它对于检测微弱的图像信号(即 $f(m, n)$ 及其四邻灰度值都很小时)非常有效，因为即使图像信号本身很弱，但比值不会很小。

5. 检测算子性能比较

利用上述算子分别对图 7.3(a) 所示的 Car 图像进行边缘检测，图 7.5(a)～(d) 分别是 Robert 算子、Sobel 算子、Prewitt 算子和 Kirsch 算子的检测结果。

7.2.3 Canny 边缘检测算子

Canny 于 1986 年提出了基于最优化算法的边缘检测算子，具有良好的信噪比和检测精度。近年来，该算子在许多图像处理领域中得到广泛的应用。根



(a) Robert 算子　　(b) Sobel 算子

(c) Prewitt 算子　　(d) Kirsch 算子

图 7.5　边缘检测算子的检测结果

据边缘检测的有效性和定位的可靠性，Canny 对最优边缘检测器所需的特性进行了研究，推导出最优检测器的数学表达式。对于不同类型的边缘，Canny 边缘检测算子的最优形式是不同的。Canny 推导得到的一维边缘检测器与一维 M－H 边缘检测器几乎是一样的，这是由于一阶导数算子的最大值和 M－H 算子的零交叉是一致的。Canny 算子在二维情况下具有良好的方向性，因此它的边缘检测和定位性能比M－H算子要好，具有更好的边缘强度估计，并且能产生边缘梯度方向和强度两个方面的信息。

1. Canny 边缘检测算子的基本原理[5]

Canny 把边缘检测问题转化为检测单位函数极大值的问题。在高斯噪声中，一个典型的边缘代表一个阶跃的强度变化，根据这个模型，Canny 给出了评价边缘检测性能优劣的三个指标，即信噪比、定位精度和单边缘响应。

(1) 信噪比指标。信噪比的数学表达式为：

$$\mathrm{SNR}=\frac{\left|\int_{-W}^{+W}G(-x)h(x)\mathrm{d}x\right|}{\sigma\sqrt{\int_{-W}^{+W}h^{2}(x)\mathrm{d}x}}$$

式中：$h(x)$是窗宽为 $2W$ 的滤波器的冲击响应函数；$G(x)$代表边缘函数；σ 是高斯噪声的均方差，信噪比越大，则所提取的边缘质量越好。

（2）定位精度指标。定位精度的数学表达式为：

$$\mathrm{Localization}=\frac{\left|\int_{-W}^{+W}G'(-x)h'(x)\mathrm{d}x\right|}{\sigma\sqrt{\int_{-W}^{+W}h'^{2}(x)\mathrm{d}x}}$$

式中：$G'(x)$和 $h'(x)$是 $G(x)$和 $h(x)$的导数；Localization 越大则定位精度越高。

（3）单边缘响应准则。要保证单边缘只有一个像素响应，则要求在 h 对噪声的响应中，两个相邻最大值间的距离为 $D_{max}(h)$，检测算子的脉冲响应导数 h'的零交叉点平均距离 $D_{zc}(h')$应满足公式：$D_{max}(h)=2D_{zc}(h')=kW$。

式中：$k<1$，$D_{zc}(h')=\pi\left\{\frac{\int_{-W}^{+W}h'^{2}(x)\mathrm{d}x}{\int_{-W}^{+W}h''(x)\mathrm{d}x}\right\}^{1/2}$。

Canny 首次将上述判据用数学的形式表达出来，然后采用最优化数值方法，得到对应给定边缘模型的最佳边缘检测模板。对于二维图像，需要使用若干方向的模板分别对图像进行卷积处理，再取最可能的边缘方向。Canny 的分析是针对一维边缘中的阶跃型边缘，推导出的最优边缘检测器的形状与高斯函数的一阶导数类似，利用二维高斯函数的对称性和可分解性，可以很容易计算高斯函数在任意方向上的方向导数与图像的卷积。所以在实际运用中选取高斯函数的一阶导数作为阶跃边缘的次最优检测算子。下面推导二维次最优阶跃边缘检测算子的数学表达式。设二维高斯(Gaussian)函数为

$$G(x,y)=\frac{1}{2\pi\sigma^{2}}\exp\left(-\frac{x^{2}+y^{2}}{2\sigma^{2}}\right)\tag{7.2-21}$$

在某个方向 $\boldsymbol{n}$ 上$G(x,y)$的一阶方向导数为

$$G_{n}=\frac{\partial G}{\partial n}=\boldsymbol{n}^{\mathrm{T}}\nabla\boldsymbol{G}\tag{7.2-22}$$

式中：$\boldsymbol{n}=\begin{bmatrix}\cos\theta\\ \sin\theta\end{bmatrix}$，$\nabla\boldsymbol{G}=\begin{bmatrix}\partial G/\partial x\\ \partial G/\partial y\end{bmatrix}$，$\boldsymbol{n}$ 是方向矢量，$\nabla\boldsymbol{G}$ 是梯度矢量。将图像

$f(x, y)$与G_n作卷积,同时改变$\boldsymbol{n}$的方向,当$G_n * f(x, y)$最大时,则$\boldsymbol{n}$正交于检测边缘的方向。由$\frac{\partial(G_n * f(x, y))}{\partial \boldsymbol{n}} = 0$得到方向$\bar{n} = \nabla \boldsymbol{G} * f(x, y) / |\nabla \boldsymbol{G} * f(x, y)|$,此时有最大输出响应

$$\begin{aligned} |G_n * f| &= |\cos\theta(\partial G/\partial x) * f(x, y) + \sin\theta(\partial G/\partial y) * f(x, y)| \\ &= |\nabla \boldsymbol{G} * f(x, y)| \end{aligned} \tag{7.2-23}$$

以上为Canny二维次最优阶跃边缘检测算子的数学推导。

2. Canny边缘检测算法

根据Canny的定义,边缘点的强度可以由$|G_n * f|$来估计,那么中心边缘点的强度为算子G_n与图像$f(x, y)$的卷积在梯度方向上的最大值,这样就可以根据在每一点的梯度方向上,判断此点强度是否为其邻域的最大值,来确定该点是否为边缘点。根据以上的分析,利用Canny算子检测边缘的具体算法如下:

(1) 用高斯滤波器对图像滤波,以消除图像中的噪声。例如方差为1.4的高斯函数近似模板可为:

2	4	5	4	2
4	9	12	9	4
5	12	15	12	5
4	9	12	9	4
2	4	5	4	2

但高斯滤波器的方差和模板大小应该根据图像中所含的目标情况来适当选择。

(2) 对滤波后图像中的每个像素,计算其梯度幅值A和方向ϕ,可以采用7.2.2小节中的梯度算子的水平模板P和垂直模板Q作为计算x方向和y方向梯度分量的计算模板。根据式(7.2-2)和式(7.2-3)得到图像$f(x, y)$上点(i, j)处的梯度幅值$A(i, j)$和方向$\phi(i, j)$。

(3) 对梯度幅值进行"非极大抑制"。$A(i, j)$的值越大,其对应的图像梯度值也越大。但只从图像$f(x, y)$上点(i, j)处的梯度幅值$A(i, j)$的大小并不能确定该点是否为边缘点。为了更好地确定边缘,必须细化幅值图像中的屋脊带,这样才会生成细化的边缘,保留幅值局部变化最大的点,该过程采用"非极大抑制"来实现。

该算法首先对图像的梯度方向进行方向角规范化。图像中每个像素只有4个

可能的方向与邻点相连：0°（水平方向0），45°（正对角线1），90°（垂直方向2）和135°（负对角线3）。如图7.6所示。边缘方向必是这4种中最接近的一种。因此方向角被规范到以下4个角度：

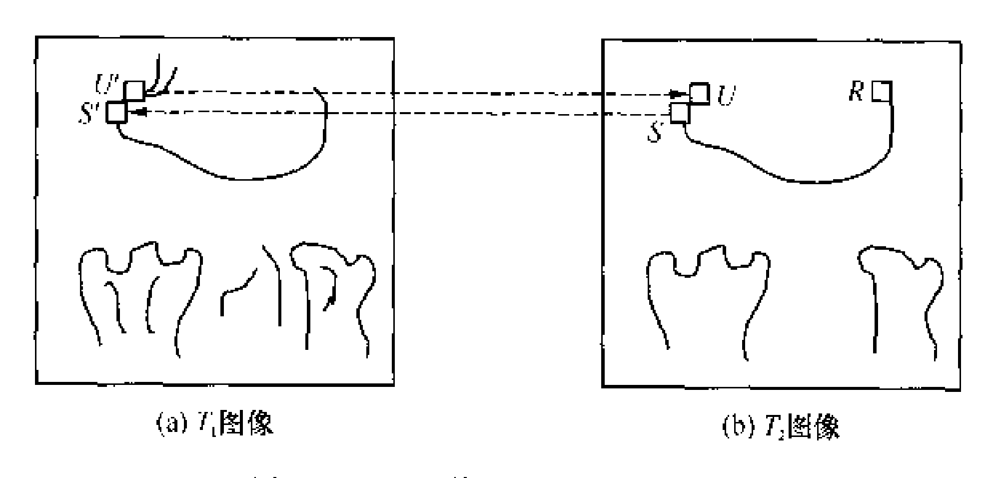
0°：0°～22.5°，157.5°～180°；45°：22.5°～67.5°；90°：67.5°～112.5°；135°：112.5°～157.5°。

3	2	1
0	x	0
1	2	3

图7.6 非极大抑制的梯度方向划分

"非极大抑制"通过抑制梯度方向上所有非屋脊峰值的幅值来细化$A(i,j)$中的梯度幅度屋脊。将图7.6所示3×3方向模板作用于幅值阵列$A(i,j)$的所有点，在每一个点上，中心像素与沿着梯度线的两个像素进行比较，若在中心点处的幅值$A(i,j)$不比沿梯度线方向上的两个相邻点幅值大，则将$A(i,j)$置为零。将那些不是极大值的点置为零，这一过程即为非极大值抑制。经过处理后宽屋脊被细化为只有单像素点宽，在非极大值抑制过程中，保留了屋脊的高度值。

(4) 阈值化和边缘连接。经过"非极大值抑制"得到的结果是一个图像的边缘阵列，但仍然存在许多由噪声和纹理引起的假边缘，需要进一步阈值化处理，去除假边缘。如果用单阈值处理的话，阈值的选择比较困难，阈值选择过大会把一些边缘信息去除，而阈值选择过小又会保留部分假边缘。一种有效的解决方法是选用两个阈值。设非极大值抑制图像为$N(i,j)$，有阈值τ_1和τ_2，且$\tau_2>\tau_1$。用这两个阈值处理边缘图像$N(i,j)$，把梯度值小于τ_1的像素的灰度设为0，得到边缘图像T_1，同理，得到T_2。由于T_2是用高阈值得到的，因此，它含有较少的假边缘，但也可能损失了一些边缘信息；而T_1的阈值较小，保留了较多的信息。把图像T_2作为基础，以T_1为补充连接图像的边缘。具体过程如图7.7所示，(a)为T_1图像，(b)为T_2图像。



(a) T_1图像 (b) T_2图像

图7.7 Canny算法的边缘连接示意图

首先在图像 T_2 中扫描，一旦遇到一个非零灰度的像素 R，跟踪以 R 为始点的轮廓线，直到该线的终点 S。接着在图像 T_1 中比较与图像 T_2 中 S 点位置对应的 S' 的 8-邻域。如果在 S' 点的 8-邻域有非零像素 U' 存在，则将其包括到图像 T_2 中，作为 U。同理，重复在 T_2 中继续寻找跟踪以 U 为开始点的轮廓线。这样循环直到在图像 T_1 和 T_2 中都无法继续为止。包含 R 的轮廓线的连接已经完成，可标记为已访问过。然后依次可以重复寻找图像中的每一条边缘线，直到在图像 T_2 中再也找不到新的轮廓线为止。

3. Canny 算子检测性能

利用 Canny 算子对图 7.3(a)所示的 Car 图像进行边缘检测，高斯滤波器采用方差为 1.4 的 5×5 模板，利用 Robert 算子计算梯度的幅值和方向，并且双阈值分别为 $\tau_1=0.8$，$\tau_2=0.9$，检测结果如图 7.8 所示。注意若选用了不当的高斯滤波器、梯度算子或者阈值，则有可能得到不理想的边缘检测结果。

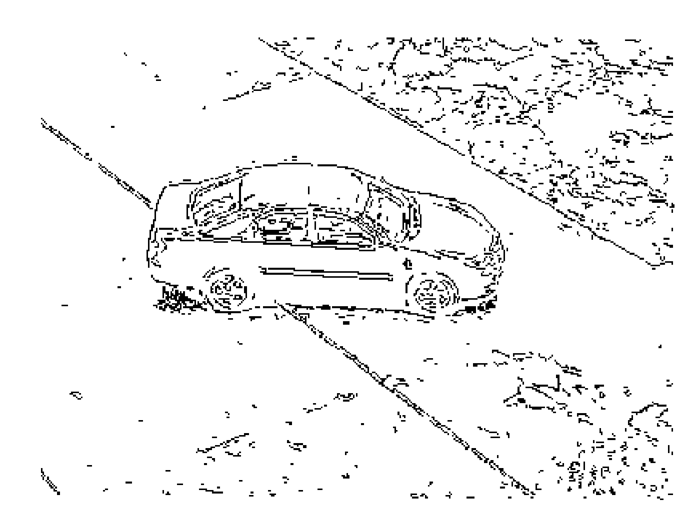


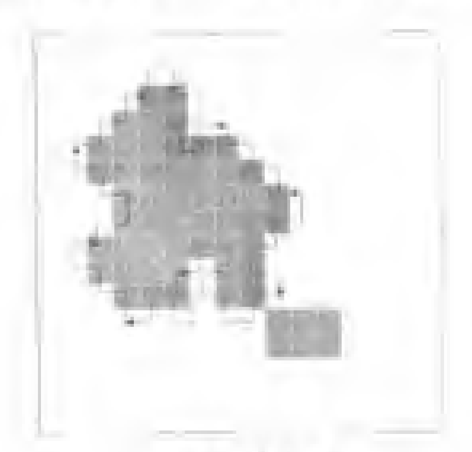
图 7.8 Canny 算子的检测性能

7.2.4 轮廓跟踪

轮廓跟踪(Boundary Tracking)方法就是从图像(或梯度图)中的一个边缘点开始，按照一定的准则搜索并连接相邻边缘点从而检测出边缘。新的边缘点的确定不仅依赖于该点的局部特性，还依赖于先前已经确定了的边缘点，这样可以从一定程度上消除噪声的影响。一般的边缘跟踪算法主要包括三个步骤：(1)确定起始点。起始点的选择很重要，通常影响到检测结果。(2)确定一定的准则，在已有的边缘点基础上确定新的边缘点或更新边缘点。(3)确定终止条件，当满足终止条件时，过程停止。下面主要介绍两种轮廓跟踪算法：虫随法和 Snake 算法。这里把 Snake 算法也归为轮廓跟踪类型，主要是由于 Snake 算法具有轮廓跟踪的特点，只不过其初始状态是它具有初始点集(初始曲线)。

1. 虫随法

这里介绍一种简单实用的轮廓跟踪算法，称为虫随法[illegible]。在图7.9的二值图中设想有一只虫从白像素区向黑像素区爬行，当虫穿过黑像素时便左转弯，并走向下一个像素，同时，始终遵循遇黑左转、遇白右转的原则，一直到虫爬回原来的出发点为止。虫的爬行轨迹反映了物体的边缘轮廓。在图7.9(a)中目标右下角的突出像素与邻像素是8连通的，它没有包括在边界内，而在图7.9(b)中，在改变小虫出发点后这一突出像素被包括在边界之内了。因此，采用这种方法找边界，所得结果与出发点有关(当然二者的差别是很小的)。


(a) 出发点在左上角

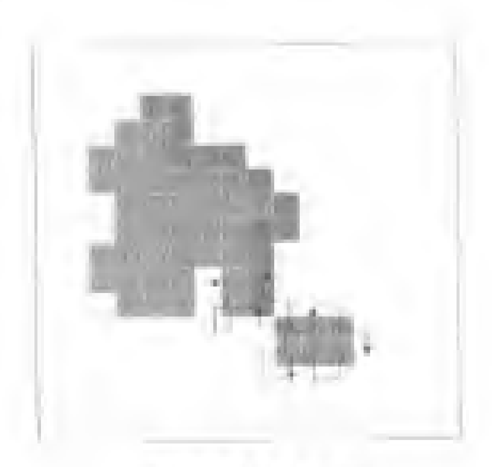

(b) 出发点在右下角

图7.9 虫随法示例

2. Snake算法

传统的边缘检测方法被广泛地认为是自底向上的过程。其中，MIT人工智能实验室的Marr的分层计算理论产生了广泛的影响。他认为需要3个独立的层次来表达视觉信息的处理过程，计算只能依赖于从图像本身获得的信息进行，不可能使用高层的信息。基于这种分层理论，人们提出了许多算法，但轮廓提取问题仍然没有得到满意的解决。其中一个主要的原因就是底层的误差传播到了高层，而没有修正机会。

Kass等人认为在许多图像理解任务中，底层事件的正确理解依赖于高层知识。基于这样一种思想，Kass等人在1987年第一届国际视觉会议上提出了称为Snake的主动轮廓线模型(Active Contour Model)[illegible]。Snake模型的引人之处在于，它对于范围广泛的一系列视觉问题给出了统一的解决方法。该模型已经被许多学者应用于边缘提取、图像分割和运动跟踪等领域。

(1) Snake模型的原理。Snake模型，简言之就是表征拟合误差的"能量"为最小化的曲线，其原理为：设对于拟合目标有一个待选曲线集，定义能量函数与待选集中每一条曲线相关联，能量函数的设计原则就是：有利属性要能导致能量减小，

这里的有利属性包括：曲线连续、平滑、曲线与高梯度区域接近以及其他一些具体的先验知识。当活动轮廓在取值范围内移动时，就能在能量函数的指导下收敛到局部边界，且能保持曲线的连续和平滑。

Snake是能量极小化的样条，在曲线本身的内力和图像等外部约束力作用下移动的变形轮廓线。内力和外力的作用是不同的：内力起平滑约束作用，外力则引导Snake模型向图像特征移动。打个比方，设想Snake模型是由两种弹性材料构成：弦与杆。前者使轮廓抵抗韧性，而后者使轮廓抵抗弯曲。两者的共同作用就构成了Snake模型的内力，因此，内力决定着Snake模型的灵敏性。施加在Snake模型上的外力来自图像或更高层的处理，外力将Snake模型推离不期望的特性，引向图像特征。因此Snake模型是"主动"的轮廓线模型，它锁定在图像特征附近，准确地将它们极小化。

(2) Snake的数学模型。设用可变形曲线 $v(s) = \{x(s), y(s)\}$ 来定义Snake模型曲线，其中 $s \in [0, 1]$，s 是归一化的曲线长度。Snake模型的总能量函数为：

$$E(v) = \int [E_{int}(v(s)) + E_{ext}(v(s))] ds \tag{7.2-24}$$

其中，E_{int}是内部能量，控制Snake模型的特性，其定义为：

$$E_{int} = \frac{1}{2}(\alpha \mid v'(s) \mid^2 + \beta \mid v''(s) \mid^2) \tag{7.2-25}$$

$v'(s)$和$v''(s)$分别是 v 对 s 的一阶和二阶导数，其一阶导数取定轮廓的连续性，二阶导数确定轮廓的曲率或角点。系数 α 和 β 分别控制Snake模型的弹性和刚性。Snake模型对轮廓的灵敏性就依赖于这两个系数。这些参数操纵着模型的物理行为和局部连续性。当 $\alpha|v'(s)|^2$ 取极小值时，Snake拉伸受到限制；当 $\beta|v''(s)|^2$ 取极小值时，Snake弯曲受到限制。

外部能量 E_{ext}引导Snake变形曲线收敛于特定的图像特征。E_{ext}一般由图像特征决定。由于这些图像特征只能根据特定的问题定义，因此外部能量 E_{ext} 一般不易确定，没有统一的数学表达式，必须从问题本身的特性出发，进行分析定义。外部能量 E_{ext}一般描述为：

$$E_{ext} = E_{image} + E_{con} \tag{7.2-26}$$

其中，E_{image}为图像能量，可描述为

$$E_{image} = \varepsilon_1 E_{gray} + \varepsilon_2 E_{edge} + \varepsilon_3 E_{line} \tag{7.2-27}$$

式中：E_{gray}，E_{edge}，E_{line} 分别是图像的灰度、边缘和线的能量；ε_1，ε_2，ε_3 是其权值。E_{gray}，E_{edge}，E_{line}可分别定义为：

$$\begin{cases} E_{grav}(v(s)) = |f(v(s))|^2 \\ E_{edge}(v(s)) = |f'(v(s))|^2 \\ E_{line}(v(s)) = |f''(v(s))|^2 \end{cases} \tag{7.2-28}$$

式中：$f(v(s))$是图像在点$v(s)$处的灰度值。Snake在类似重力的图像力的作用下趋向于边缘特征明显的区域。

外部约束力能量E_{con}是人为赋予Snake的，根据经验引入。它可使Snake向预定目标方向运动，使轮廓收敛到图像不同的特征点。例如Kass把E_{con}定义为火山模型

$$E_{con} = -k_v |v(x, y) - Volcano(x, y)|^2/2,$$

其中$Volcano(x, y)$表示火山，k_v为火山能量系数。

设$V = (v_1, v_2, \cdots, v_n)$为离散化的轮廓曲线，边缘检测问题转化为求最优轮廓曲线$V' = \{v'_1, v'_2, \cdots, v'_n\}$，满足：

$$E_{snake}(V') = \min_V(E_{snake}(V)) \tag{7.2-29}$$

(3) Snake模型的基本算法。下面给出Snake模型算法的基本步骤：

① 用特定机制定义Snake初始轮廓，如：可用人机交互方法来确定初始轮廓。然后定义轮廓起始点。下面步骤将从该起始点开始顺时针逐个轮廓点计算Snake能量值，以确定最小能量。

② 用当前轮廓点v_i构造内部能量，在离散的情况下，内能E_{int}近似为

$$E_{int}(v_i) = (\alpha_i \| v_i - v_{i-1} \|^2 + \beta_i \| 2v_i - v_{i-1} - v_{i+1} \|^2)/2。 \tag{7.2-30}$$

按照某种规则构造轮廓点v_i处的外能，通常采用图像梯度幅值来构造Snake的外能：

$$E_{ext} = -\gamma \| \nabla f(v_i) \|^2 \tag{7.2-31}$$

③ 计算此时的Snake能量。

④ 将该轮廓点变动为右侧邻接点，重复步骤②～③，然后在其8邻域内顺时针寻找最小Snake能量值的位置，确定该位置点后，把该点记为新轮廓点，然后继续下一个轮廓点。

⑤ 完成所有轮廓点的计算后，重复步骤②～④，迭代寻找最佳结果，直到模型收敛或达到迭代最大次数。

(4) Snake模型的性能及其改进。采用式(7.2-30)和式(7.2-31)作为Snake模型的内能公式和外能公式，对图7.3(a)的Car图像进行处理，图7.10(a)给出了Snake的初始轮廓，最终的收敛结果如图7.10(c)。





(a) 初始轮廓



(b) 收敛过程中



(c) 最终结果

图 7.10 Snake 算法

通过对 Snake 算法的分析和边缘检测结果的比较，Snake 模型具有一些经典方法所无法比拟的优点：

① 图像数据、初始估计、目标轮廓及基于知识的约束统一于一个特征提取过程中；

② 经适当地初始化后，它能够自主地收敛于能量极小值状态；

③ 尺度空间中由粗到精的极小化能量可以极大地扩展捕获区域和降低计算复杂性；

④ Snake 模型具有一定的稳定性和简易性。

但同时，Snake 模型也有其自身的缺点：

① 对初始轮廓非常敏感，也就是 Snake 模型的初始位置必须接近图像期望的边缘；

② Snake 模型不一定能够收敛到全局极值，特别是在灰度图像里出现凹度或拐弯处；

③ Snake 模型在能量极小化过程中的收敛速度太慢或数值不稳定等。

针对 Snake 模型的上述问题，许多学者提出了一些改进方法，以改进 Snake 的

性能，如：Cohen 等人提出的主动轮廓线的"气球"模型[1]，Xu 等人提出的基于梯度矢量流- Gradient Vector Flow(GVF)的 Snake 模型[2]。关于这些改进方法的详细情况，可参阅相应的参考文献。

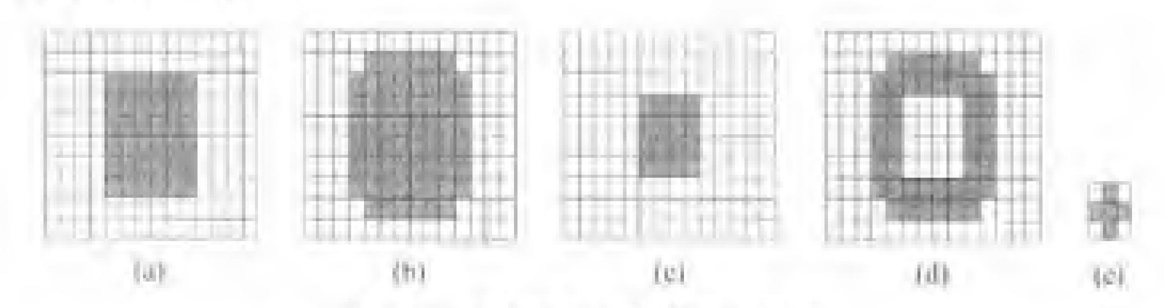
7.2.5 边缘的形态检测

数学形态学的数学基础和所用的语言是集合论，其基本运算主要有四个：膨胀、腐蚀、开启和闭合。基于这些基本运算可推导和组合许多实用的数学形态学实用算法。基于数学形态学的算法具有天然的并行实现结构。它应用在边缘检测，可以实现除去不相干的结构，并能保持图像的基本形状特性。下面主要描述数学形态学在边缘检测方面的应用，关于形态学知识参看附录2。

前面我们介绍了很多边缘检测算子进行边缘的检测，这类算子一般对噪声很敏感并且通常在检测边缘的同时增强了噪声。而在用形态梯度进行形态边缘检测时，则不会加强噪声，但仍对噪声比较敏感。形态梯度[13][14]有多种定义形式。设 f 表示二值图像，B 表示结构元素，这里 f 和 B 都是集合，$\oplus$代表膨胀运算，Θ 代表腐蚀运算。一个基本的形态梯度 G 定义为：

$$G=(f\oplus B)-(f\Theta B) \tag{7.2-32}$$

式(7.2-32)的示例如图7.11。图7.11(a)为图像 f，(e)为结构元素，(b)为 $f\oplus B$，(c)为 $f\Theta B$，(d)为 G。



(a) (b) (c) (d) (e)

图7.11 形态梯度 G 的计算过程示例

从图7.11(d)可以看出，G 给出的边界具有两个像素宽度。为了得到单像素宽度的边界，重新定义两个形态梯度 G_1 和 G_2。

$$G_1=(f\oplus B)-f \tag{7.2-33}$$

$$G_2=f-(f\Theta B) \tag{7.2-34}$$

7.2.6 霍夫(Hough)变换

霍夫变换是在预先知道区域形状的条件下，可以方便地检测边界曲线的一种

常见方法。本节对该变换作详细的描述。

1. 霍夫变换的基本原理

霍夫变换是一种快速形状匹配技术，利用图像全局特性而直接检测目标轮廓。霍夫变换的基本思想是通过在参数空间确定参考点来检测图像空间的目标。它对图像进行某种形式的坐标变换，使得经过变换后原图上给定形状的曲线上的所有点都集中到变换空间某些位置上，形成峰点。这样，就把对原图中给定形状曲线的检测问题转化为找变换空间中峰点的问题。经典的霍夫变换适用于曲线形状可以用参数来描述，而曲线位置未知的情况下检测曲线。下面通过两个例子说明霍夫变换的检测过程。

例 1 过给定点 P 的直线的检测问题。

有一幅白背景上有一些黑色线条和点的数字图像，经过二值化后，若以 0 表示白背景上的像素点，以 1 表示黑线条或黑点处的像素，要求找一种算法能检出过 P 点的任意方向的直线段。

若把 P 点作为原点，则过 P 点的直线方程为 $y = mx$。在图像空间中，设 P_i 的坐标为 (x_i, y_i)。现在造一个一维数组 A，数组的下标对应于 P_i 的点与原点连线和水平方向间的夹角 θ。若每隔 1 度对应于一个分量，数组 A 应有 360 个分量。数组 A 的每个分量先全部置 0，然后对图像空间中每个值为 1 的点从 P_1 到 P_n 逐点计算它们的 θ 角，并把与这个 θ 角相对应的数组分量的值加 1。图 7.12 是一个图例，该图中有 5 个点都处于夹角为 45°的线上，则经过上述变换过程后，数组 A 在下标 45 的位置上有一个峰值 5，而其他不共线的点散布在数组 A 中形不成峰点。本例中一维数组 A 相当于一个一维的变换空间，通过上述由原始图像到参数空间的变换，把直线检测问题转化为找参数空间中峰点的问题了。

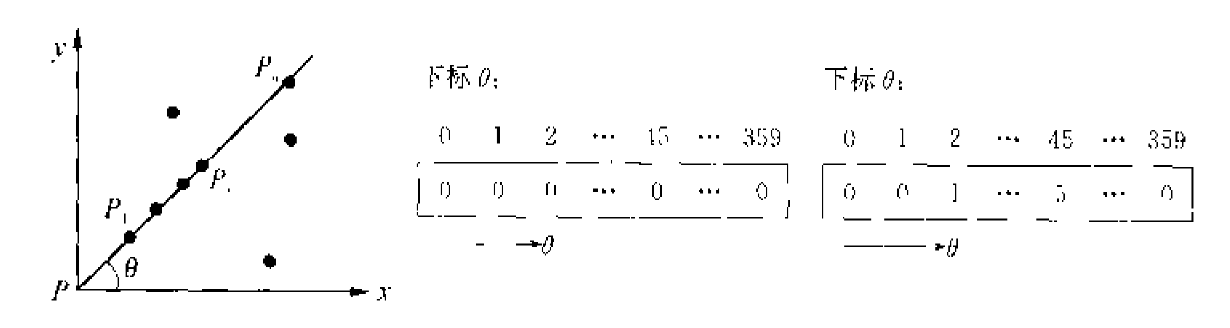


图 7.12 过原点直线的检测

例 2 任意方向和任意位置直线的检测问题。

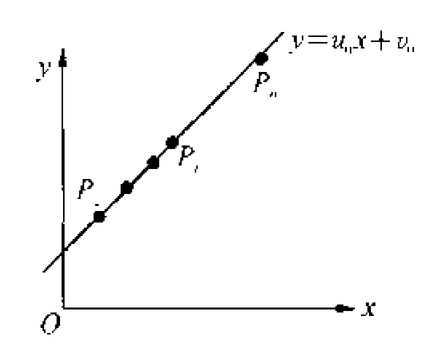
现在要用到二维的变换空间 (u, v)。在原始图像空间 (x, y)，直线方程为

$$y = ux + v \tag{7.2-35}$$

式中：u 为斜率，v 为截距。而对于这个直线上的一个点 $P_i = (x_i, y_i)$ 来说，它在

变换空间中应该满足方程式 $v=-x_iu+y_i$。

也就是说，图像空间中的一个点 (x_i, y_i) 对应于变换空间 (u, v) 中的一条直线；而变换空间中的一个点 (u_0, v_0) 则对应于图像空间中的一条直线 $y=u_0x+v_0$，该直线斜率为 u_0，截距为 v_0。既然图像空间中一个点对应于变换空间中的一条直线，为什么图像空间中的一条直线反而只对应于变换空间中的一个点，而不是对应于更多的东西呢？可以这样理解：试用一张白纸来代表变换空间，对于图像空间中位于直线上的每一个点，在代表变换空间的纸上用淡铅笔画上一条与之对应的直线，由在图像空间中共直线的所有点所对应的铅笔线应相交于一点，这一点上的铅笔痕最深。这个交点在上述意义下和变换空间中的一条直线相对应。如果用二维数组代替上述的白纸，用数组中位于对应线上的元素的值加1来代表上述划淡痕的过程，则纸上笔痕最深的点相应于二维数组中的峰点。图7.13是上述情况的示意图。一般情况下，若变换空间是二维空间，则图像空间中一个点与变换空间中的一条线（不一定是直线）相对应。



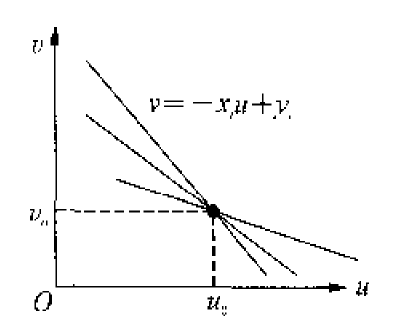


图7.13 任意方向和任意位置直线的检测

通过上述两例，可以得到Hough变换的一般步骤：

(1) 在合适的范围中量化变换空间 (u, v)。

(2) 构造一个累加数组 $A(u, v)$，它的每一个元素的下标对应于变换空间中各点的位置，其元素值表示通过该点的曲线的条数。刚开始时先将 $A(u, v)$ 的各元素置0。

(3) 若数字图像中像素值为"0"表示背景点，为"1"表示待检测共线的像素点，则对图像空间中每个值为"1"的点在变换空间中找到与其对应的曲线，并将该对应曲线所在位置上的所有点处的累加数组元素的值加1。

(4) 找累加阵中的峰点。峰点的位置对应于图像空间中待检测线的位置，峰的高低反映了这条线上像素点的多少。

Hough变换的一个突出优点是抗干扰能力强。如果待检测线条上有小的扰动或断裂，甚至是对虚线的检测，进行Hough变换后，在变换空间中仍能得到明显的峰点。

变换的具体形式可以不同。上述例 1 是例 2 的特殊情况，原则上若采用例 2 中的二维变换，然后找 $v=0$ 即横轴上的峰点也能解决问题，但若采用例 1 中所用的一维变换空间，计算量显然小于二维变换空间。在例 2 的情况，为了检测任意方向和位置的直线也可以采用另一种变换方式，即不用斜率-截距作为变换空间而采用极坐标(r, θ)作为变换空间，也就是把式(7.2-35)改为

$$x_i\cos\theta + y_i\sin\theta = r \tag{7.2-36}$$

则(x, y)空间中的一点(x_i, y_i)对应于(r, θ)空间中的一条正弦曲线，如图 7.14 所示。

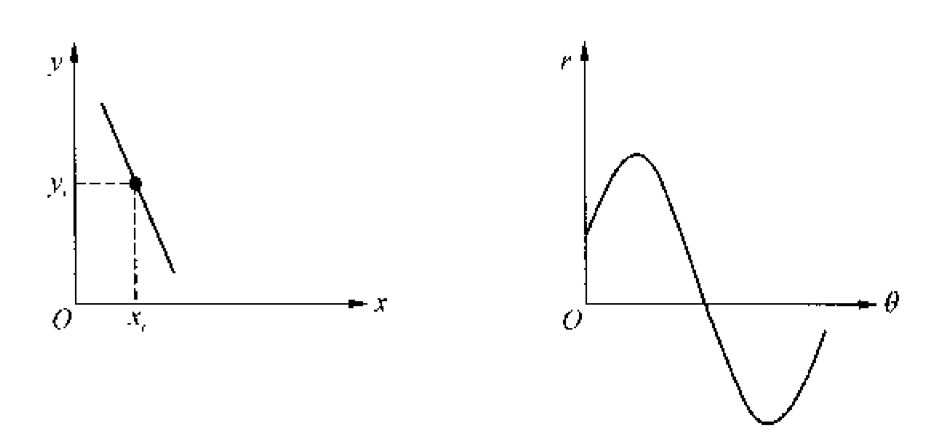


图 7.14 极坐标(r, θ)作为变换空间的直线检测

而变换空间中，这条正弦曲线上的任一点对应于(x, y)空间中的一条直线。这条直线必通过(x_i, y_i)这个点，而(x, y)空间中共线的点经过变换后所对应的各正弦曲线都相交于一点。交点的(r, θ)值与(x, y)空间中直线位置的关系如图 7.15 所示。如果就解决直线检测问题本身而言，后一种变换较前一种好，因为它避免了垂直线的无限大斜率问题。

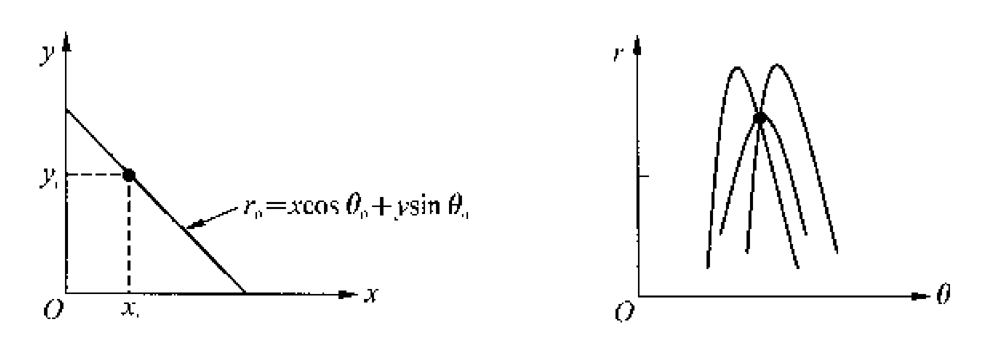


图 7.15 (r, θ)值与(x, y)空间中直线位置的关系

在变换时利用梯度方向信息可以减少计算量，提高检测的可靠性。但是在实现时，必须注意梯度方向估计的精度不高，会造成峰点的模糊，有兴趣的读者可参阅文献[14]。

Hough 变换不仅可以用来检测直线，也可方便地用来检测椭圆、圆和抛物线等形状的曲线，检测的基本原理不变。现以检测位置未知、半径 r 已知的圆为例来简要说明。在(x, y)空间中，圆的方程为

$$(x-a)^2+(y-b)^2=r^2 \tag{7.2-37}$$

现在 x、y 是变量，a、b 为未知参量，r 为已知参量，取(a, b)作为变换空间。在(x, y)空间中，圆上任一点 $P_i=(x_i, y_i)$ 与(a, b)空间中的圆

$$(a-x_i)^2+(b-y_i)^2=r^2 \tag{7.2-38}$$

相对应。(x, y)空间中所有共圆的点 P_i，例如 $(x-a_0)^2+(y-b_0)^2=r^2$ 的点，在(a, b)空间的对应曲线相交于一点(a_0, b_0)。于是用上述 Hough 变换的算法，对每个 P_i，使(a, b)空间中与式(7.2-38)对应位置上所有累加阵的分量加 1，最终可以由(a_0, b_0)点出现的峰点，检测到在(x, y)平面上有以(a_0, b_0)为中心、半径为 r 的圆存在。

2. 广义霍夫变换

在上面的例子中曲线都可以用解析式来表示。但当所需检测的曲线或区域轮廓不能或不易用解析式描述时，应对原有算法进行改进，改进后的算法称为广义霍夫变换[36]。其基本思想不变，只是采用了表格来建立曲线或轮廓与参考点间的关系。该算法要求已知图像上曲线或区域轮廓的形状、大小及方向，需要检测其位置信息。下面介绍该算法。

第一步，对待检曲线或轮廓，利用表格建立参考点与轮廓点的联系。先在所给轮廓内部取定一个参考点(x_r, y_r)。对任意一个轮廓点(x, y)，从(x, y)到(x_r, y_r)的矢量为 $\boldsymbol{r}$。计算点(x, y)处的梯度方向，设其方向角为 Φ。如图 7.16 所示。这样，每个轮廓点对应一个梯度角 Φ（反过来一个 Φ 可能对应多个轮廓点）。把参考点和这个轮廓点的相对位置存储起来，即以梯度的方向 Φ 为下标，存储 $\boldsymbol{r}$ 的方向 $\alpha(\Phi)$ 和长度 $r(\Phi)$。参考点的坐标与轮廓点的坐标关系：

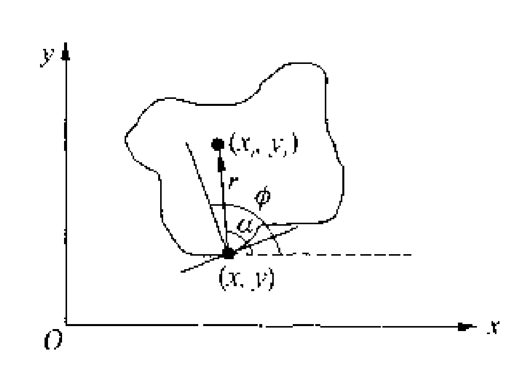


图 7.16　任意形状边界的广义霍夫变换

$$\begin{cases} x_r = x - r(\Phi)\cos[\alpha(\Phi)] \\ y_r = y + r(\Phi)\sin[\alpha(\Phi)] \end{cases} \tag{7.2-39}$$

据此，以 Φ 为自变量，得到一个表征给定形状的表——R 表。R 表本身与轮廓的绝对坐标无关，只是描述了轮廓形状。设轮廓上共有 n 个点，则 R 表的格式如表7.1

所示。

表 7.1　R 表的示例(其中 $\Delta\Phi$ 为梯度角度增量)

梯度角 Φ	矢径 $r(\Phi)$	矢角 $\alpha(\Phi)$
$\Delta\Phi$	$r_1^1, r_1^2, \cdots, r_1^{n1}$	$\alpha_1^1, \alpha_1^2, \cdots, \alpha_1^{n1}$
$2\Delta\Phi$	$r_2^1, r_2^2, \cdots, r_2^{n2}$	$\alpha_2^1, \alpha_2^2, \cdots, \alpha_2^{n2}$
…	…	…
$k\Delta\Phi$	$r_k^1, r_k^2, \cdots, r_k^{nk}$	$\alpha_k^1, \alpha_k^2, \cdots, \alpha_k^{nk}$
…	…	…
$m\Delta\Phi$	$r_m^1, r_m^2, \cdots, r_m^{nm}$	$\alpha_m^1, \alpha_m^2, \cdots, \alpha_m^{nm}$

表中 r 和 α 下标分别与 Φ 一一对应，而它们的上标表示具有相同 Φ 的轮廓点。对每个边界点先计算其梯度方向 Φ，然后用 Φ 作下标在 R 表查得 α 和 r，这样可以方便地确定参考点的位置。如果待测形状和已知形状一致，对轮廓上每点用上述方法求得的参考点位置应该相同，从而在累加阵中出现峰点；反之，若待测形状和已知形状不同，则对于待测形状边界上各点，通过查 R 表算得的参考点会很分散，不能形成峰点。由此可见，任意形状边界的 Hough 边界检测算法可总结如下：

(1) 由已知形状的曲线作 R 表；

(2) 生成一个参考点累加器阵列 $A(x_r, y_r)$，并清零。

(3) 对每一个可能的边界点进行梯度角 Φ 计算，并计算可能的参考点，即对 Φ 表示的 m 种可能的状态的每一项进行计算，由式(7.2-39)算出可能的参考点。据此对累加器阵列 $A(x_r, y_r)$ 进行累加。

(4) 在对每个可能的边界点执行③后，在阵列 $A(x_r, y_r)$ 中通过找峰值确定形状的可能的位置。

以上的过程是在区域形状的旋转情况不存在和尺度不变情况下进行的，如果区域形状存在旋转 β 和比例尺度变化 s，则参数空间从 2-D 增到 4-D，但广义霍夫变换的基本方法不变，这时只需把累加阵变为 $A(x_r, y_r, s, \beta)$，并把式(7.2-39)变为式(7.2-40)：

$$\begin{cases} x_r = x + r(\Phi) \times s \times \cos(\alpha(\Phi) + \beta) \\ y_r = y + r(\Phi) \times s \times \sin(\alpha(\Phi) + \beta) \end{cases} \tag{7.2-40}$$

7.2.7　多尺度轮廓提取

图像边缘点实际上是局部图像的突变点(奇异点)，边缘也就是图像中具有意义

的奇异点的集合。边缘点在频域中表现为高频信号，而图像噪声也通常表现为高频信号，并且噪声的分布等信息一般是未知的，因此这造成了二者难以区分。有时会采用平滑滤波消除噪声，但这也会导致边缘的模糊，如果还存在其他的因素影响，则单一尺度的边缘检测算法就会在检测所有正确的并且满足实际要求的边缘方面存在困难。经过研究发现，采用多尺度边缘提取方法可以在很大程度上克服上述问题。

Rosenfeld 最先提出了多尺度边缘提取的思想[15]，并经过许多专家的发展，特别是 Marr，Hildretch 和 Witkin 等学者[16][17]，该思想逐步形成了一套较为完善的理论体系。该理论在图像边缘检测方面表现了很大的优势，通常小尺度的检测算子能够检测出灰度的细小变化，反映更多的边缘细节，但对噪声比较敏感；大尺度的检测算子能检测出灰度的粗变化，反映粗略的边缘轮廓，对噪声有较强的抑制。而一般的多尺度边缘检测算法是：首先分别在不同的尺度下采用边缘检测算子进行边缘提取，得到多个尺度下的边缘提取结果，然后采用一定的准则对这些结果进行综合，从而得到较为准确的边缘。在图像的多尺度轮廓提取算法中，基于小波的多尺度边缘提取算法具有代表性[18]。详细的理论工具可参阅相关文献，本节仅给出一种常用的多尺度边缘检测综合算法，该算法如下：

(1) 求出图像在各尺度 $s = 2^j (1 \leqslant j \leqslant J)$ 上可能的边缘 P_j。

(2) 对各 P_j 进行链接处理，得到在尺度 $s = 2^j (1 \leqslant j \leqslant J)$ 上的边缘 E_j，并令 $j = J$。

(3) 针对尺度 2^j 的每个边缘点(m, n)，把 P_{j-1} 在以点(m, n)为中心的 3×3 像素区域内所有可能的边缘点均标示为尺度 2^{j-1} 上的候选边缘点。

(4) 对尺度 2^{j-1} 上的所有候选边缘点进行链接处理，得到尺度 2^{j-1} 上的边缘 E_{j-1}，并令 $j = j - 1$。

(5) 重复步骤(3)和(4)，直到 $j = 1$。边缘 E_1 就是综合得到的图像边缘。

分解的尺度数 J 根据实际需要选择，在一般情况下，取 $J = 4$ 或 5。另外在边缘检测前应对原始图像进行预处理，加大边缘两侧灰度的差异。

7.3 基于区域的图像分割

图像是由具有特殊意义的不同区域构成的，每一个区域自身具有“相似”的特性，即在一个区域内部满足某种特征的一致性(Homogeneity)；而相邻的不同区域间具有不同的特性。基于区域的图像分割就是直接寻找并划分图像中各区域。该方法的实质就是把具有某种相似性质的像素连通起来，从而构成最终的分割区域。本节介绍一些经典的基于区域的分割算法。

7.3.1 阈值化分割

阈值化分割算法是图像分割中算法数量最多的一类。本节将较为详细地说明阈值化分割算法。

1. 阈值化分割的原理

阈值化分割技术是基于下列假设的：每个区域是由许多灰度值相近的像素构成的，物体和背景之间或不同物体之间的灰度值有明显的差别，这样可以通过取阈值(Threshold)来区分。待分割图像的特性愈接近于这个假设，用这个方法分割的效果就愈好。

该分割技术的基本原理是先确定一个处于图像灰度变化范围内的灰度阈值T，然后把图像中每个像素的灰度和这个阈值T相比较，并根据一定规则将像素分为两类，例如：把灰度值大于阈值的像素归为一类；像素值小于阈值的像素归为另一类。不同类的像素一般属于图像中不同的区域，这样根据阈值对像素进行分类即可达到区域分割的目的。下面通过一个例子说明该原理。

设有图像$f(x, y)$，其灰度范围是$[z_1, z_2]$，该图像对应的灰度级直方图(Histogram)如图7.17。

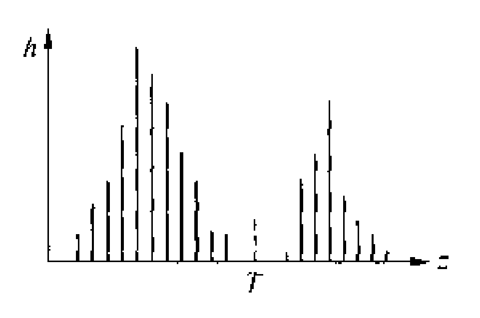


图7.17 图像的灰度级直方图

该直方图的两个峰分别对应着物体目标和背景。因此为了把各像素分成两个不同的类，需要取定一个阈值T。由于两个峰间有明显的谷底，把阈值T选在谷底处，如图7.17所示。在这里所有$f(x, y) > T$的点称为物体目标点；否则，就称为背景点。如果在分割过程中同时进行二值化，则有：

$$f_T(x, y) = \begin{cases} 1, & f(x, y) \geq T \\ 0, & f(x, y) < T \end{cases} \tag{7.3-1}$$

由此可见，阈值化分割算法可分为两个步骤：

① 根据一定的规则确定适当的分割阈值；

② 将该阈值和像素值相比较来分类像素。

在上述步骤中，确定适当的阈值是该分割算法的关键。阈值T的正确选择对于正确检测目标是十分重要的，直接关系到分割精度。下面几小节说明一些确定阈值的方法。要特别注意的是，阈值分割时只考虑了像素的本身值，未考虑像素的空间位置。因此根据阈值划分到同一类的像素可能属于图像中不连通的区域，这时，通常需要借助于其他一些方法进一步确定目标区域。

2. 阈值的确定

阈值 T 的选择是很重要的，它是阈值化分割算法的关键问题。现在已提出的阈值确定方法有很多。阈值确定函数可以写为一般形式：

$$T = T[x, y, p(x, y), f(x, y)] \tag{7.3-2}$$

式中：$f(x, y)$ 表示灰度值；$p(x, y)$ 代表某种图像性质。即，阈值 T 一般是 (x, y)、$p(x, y)$ 和 $f(x, y)$ 的函数。借助上式，阈值的确定大致可以分为三类：①仅依靠各像素值 $f(x, y)$ 的阈值。②基于区域性质 $p(x, y)$ 和像素值 $f(x, y)$ 的阈值。③基于 (x, y)、$p(x, y)$ 和 $f(x, y)$ 的阈值（动态阈值）。当然还可以有其他的确定方法，但一般可以分为以上三种。图像的直方图是图像各像素灰度值的一种统计，许多常用的阈值确定方法就是依据直方图的。近年来，一些新的技术在阈值分割算法中得到了很好的应用，例如：视觉特性、小波、模糊数学、遗传算法和信息论（熵）等。但这些新的阈值分割算法依然可以归结到以上三种方法之中，直方图依然是作为选择阈值的重要依据。下面介绍一些重要的阈值确定方法。

（1）极小阈值点。极小阈值点的方法是以图像的灰度直方图为依据的。该方法将直方图的包络看成一条曲线，则将选取直方图的谷底（阈值 T）的问题转化为求曲线极小值的问题。设有直方图 $h(z)$，则 $h(z)$ 的极值点应该满足：

$$\frac{\partial h(z)}{\partial z} = 0 \quad 且 \quad \frac{\partial^2 h(z)}{\partial z^2} > 0 \tag{7.3-3}$$

则满足上述式的灰度值 z 即可作为分割阈值 T。

该方法虽然简单，但是由于在图像中与不同区域对应的灰度值往往相互重叠，或者存在着较多的噪声，造成实际图像的直方图有很多起伏，而难以简单地用一个阈值将它分得很好。图 7.18 给出一个极端的例子，原始图像只有两个灰度级，是由白底上小黑点构成的，在背景区小黑点分布稀疏，在物体区小黑点分布稠密。虽然直方图呈明显的双峰，但选用哪个门限值都无法很好地划分物体与背景。显然，可以根据小黑点的分布密度来进行分割。现在，如果想用取阈值来进行分割，就必须在分割前先对图像进行平滑处理（实际上，许多其他的分割方法，为了取得较好的效果，也必须先对原图进行一定的平滑处理）。经过平滑处理后，背景区的点的灰度值更接近其平均灰度，小黑点被填平了，物体区的平均灰度小于背景区的平均灰度，从而使两类间的灰度交迭减小，甚至消失。虽然图示的是极端情况，对一般图像进行局部平均处理，可以使背景和物体的灰度值波动减小，从而减小灰度交迭，减小误分概率。如果原直方图因两类像素交迭而没有明显的峰点和谷点，则经过局部平均处理可使直方图的峰谷变得更分明，易于选择合适的阈值。但这个预

处理方法的缺点是平滑使边缘上的细节丢失了。为了在平均处理过程中仍能保持边缘上的细节，需要改进平滑算法。这里给出一种改进算法。其基本思想是不再简单地对所有邻域点进行灰度平均，而是选择和中心点同属一类的 k 个邻域点来平均。多次迭代后可以既减少各区灰度值的波动，又在一定程度上保持边缘信息。本方法的关键是如何判断邻域点和中心点是否同属一类。这有好多种判断法，例如：①取灰度值和中心点最接近的 k 个点进行平均。②取灰度梯度小于中心点灰度梯度的邻域点。③看一组子邻域点，若它们的灰度值方差最小，就选取这组子邻域点进行平均。

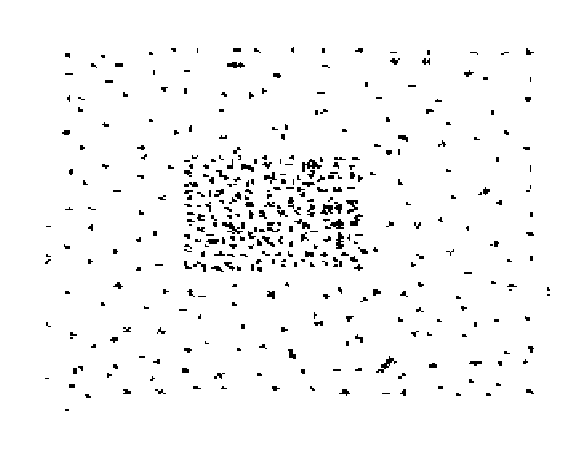


图 7.18 无法简单用取阈值来分割的图像示例

(2) 最优阈值。图像的直方图近似地表示了像素灰度值的概率分布密度函数。设图像中包含目标和背景两类区域，则直方图所代表的像素灰度值的概率分布密度实际上是目标和背景这两类的混合概率密度，即目标和背景的两个单峰分布密度函数之和。因此从统计的角度，若已知密度函数形式，那么可以找到一个最优阈值，使误分割概率最小。

设给定的图像 f 是由目标和背景两个主要的灰度级区域组成，令 z 表示灰度级值，图像的直方图 $h(z)$ 可以被看做它的概率密度函数(PDF) $p(z)$ 的一个近似。物体的灰度有概率密度 $p(z)$，而背景的灰度有概率密度 $q(z)$；并假定目标占整个画面的百分比为 θ，背景占 $1-\theta$，这两个值是目标和背景的先验概率。若某像素灰度值小于阈值 t 的点判为物体点，则把背景点错分为物体点的误分概率为

$$Q(t)=\int_0^t q(z)\mathrm{d}z \tag{7.3-4}$$

而把物体点误作为背景点的误分概率为

$$\int_t^Z p(z)\mathrm{d}z=1-\int_0^t p(z)\mathrm{d}z$$

式中，Z 是最大的灰度级。右边第二项正好是物体点的总概率 $P(t)$，即

$$P(t)=\int_0^t p(z)\mathrm{d}z \tag{7.3-5}$$

总的误分概率为

$$E(t)=(1-\theta)Q(t)+\theta(1-P(t)) \tag{7.3-6}$$

为了求最优阈值 T,使总误分概率为最小,即将上式对 t 微分,并令结果为零,即

$$(1-\theta)\frac{\mathrm{d}Q(t)}{\mathrm{d}t}-\theta\frac{\mathrm{d}P(t)}{\mathrm{d}t}=0 \tag{7.3-7}$$

或者

$$(1-\theta)q(t)-\theta p(t)=0 \tag{7.3-8}$$

可用数值解法由式(7.3-8)求出最优阈值 T。

若 $p(z)$和 $q(z)$都是正态分布,均值分别是 μ 和 v,标准差分别为 σ 和 τ。则有

$$p(t)=\frac{1}{\sqrt{2\pi}\sigma}\exp\left[-\frac{(t-\mu)^2}{2\sigma^2}\right] \tag{7.3-9}$$

$$q(t)=\frac{1}{\sqrt{2\pi}\tau}\exp\left[-\frac{(t-v)^2}{2\tau^2}\right] \tag{7.3-10}$$

代入式(7.3-8),两边取对数,得到

$$\ln(1-\theta)+\ln\sigma-\frac{(t-v)^2}{2\tau^2}=\ln\theta+\ln\tau-\frac{(t-\mu)^2}{2\sigma^2} \tag{7.3-11}$$

或

$$AT^2-BT+C=0$$

其中:

$$A=\sigma^2-\tau^2$$

$$B=2(\mu\tau^2-v\sigma^2)$$

$$C=\sigma^2v^2-\tau^2\mu^2+2\sigma^2\tau^2\ln\left[\frac{\tau}{\sigma}\cdot\frac{\theta}{1-\theta}\right]$$

解这个二次方程,选出 t_1 与 t_2 中合理的一个。当两类方差相同 $\sigma=\tau$,则得到:

$$T=\frac{1}{2}(\mu+v)+\frac{\sigma^2}{\mu-v}\ln\left[\frac{1-\theta}{\theta}\right]$$

进一步,并设 $\theta=\frac{1}{2}$,即两类先验概率相等,则得到

$$T=\frac{1}{2}(\mu+v)$$

上述讨论是在已知μ, v, σ, τ和θ之后的计算。一般来说,上述参数并不知道,可以通过直方图来估计这些参数。两类的混合分布密度为

$$H(z)=\theta p(z)+(1-\theta)q(z) \tag{7.3-12}$$

如果$p(z)$和$q(z)$采用式(7.3-9)和式(7.3-10)的正态分布,这样得到的$H(z)$是上述五个参数的函数,把它记作$H(\alpha, z)$,这里α是由μ, v, σ, τ和θ五个参数构成的五维参数向量。若实际直方图用$h(z)$表示,则为了求取这些参数,应使理论式和实际直方图之间的均方误差为最小,即使

$$D=\frac{1}{N}\sum_{i=1}^{N}[H(\alpha, z_i)-h(z_i)]^2$$

为最小,通过求它对各参数的偏微分并置为零来联立方程,则可以求得$H(\alpha, z)$的这些参数。一般来讲,确定这些参数是一件困难的事情,甚至对于高斯分布,也可能导出一组联立超越方程,通常此类方程只能通过例如共轭梯度或者牛顿法等数学求解方法解得。

(3) 基于空间信息的阈值。在实际图像处理中,图像由于多种原因而使其直方图的谷底不是很明显,对此,直方图可能呈单峰形式,在这种情况下,确定阈值比较困难。除了前面介绍的方法外,在这里介绍一种基于空间信息的阈值确定方法。该方法不仅考虑了图像像素本身的灰度值并且还利用了每个像素邻域内像素间的某些性质,例如梯度。其基本思想是利用像素邻域的一些像素间的性质扩展原始的直方图,从而形成一个新的体现这种特点的直方图,该直方图和原直方图相比,谷地更为容易确定,例如谷底更深了,因此阈值比较容易确定。有了灰度信息再加上考虑了空间信息等区域性质,则可使分割的结果更可靠、更好。但需要注意的是,上述思想的目的是为了更易确定阈值,所确定的阈值仍作用于整幅图像,因此可能对目标和背景灰度呈阶梯变化的图像效果较差。

在原有灰度直方图的基础上,除了灰度信息作为一个特征外,现在再抽取某个适当的含空间信息的特征,形成二维特征空间,从而构成二维直方图。它有两个自变量:灰度值z和某个空间信息参量(例如像素的梯度)。其因变量是像素个数。原来的一维直方图就是二维直方图在灰度轴z方向上的投影。下面介绍三种基于空间信息的阈值确定方法。

① 灰度-平均灰度法。设图像由两个灰度变动平滑的区域组成,两个区域的灰度平均值分别为z_1和z_2。每个区域内的点灰度值很接近z_1和z_2,但边界点的灰度值很可能介于z_1和z_2之间,这些值通常落在原始一维直方图的谷点附近,使峰不明显,谷不凹。设原图像f,灰度级数为L,大小为$M\times N$。采用适当的平滑模

板(例如3×3平均模板)对f进行平滑得到平滑图像$\bar{f}$,它的灰度级与大小和f相同。用f和$\bar{f}$的灰度值z和$\bar{z}$构成一个二元组。即

$$(f,\bar{f})=\{(z_{mn},\bar{z}_{mn})\}_{M\times N} \tag{7.3-13}$$

这样,用这两种灰度z和$\bar{z}$作为两个坐标轴构成一个灰度平面,从而构成一个二维直方图,在二维直方图上点$(z,\bar{z})$的函数值表示灰度为z、同时小邻域内的平均灰度为$\bar{z}$的像素的个数。

在二维直方图中,高峰主要分布在灰度平面的对角线附近,而且总体上呈现双峰。显然,这主要是因为图像中目标点和背景点所占比例最大,并且区域内部灰度波动不大,内点像素灰度和其区域平均灰度相差不大,因此对于区域内点而言基本上集中于平面的对角线附近,两个峰分别对应目标和背景。对于边界点一般其灰度值和平均值差别较大,这使得边界点对应的灰度平面点远离对角线。这样在远离平面对角线的坐标处,峰的高度急剧下降,这部分反映了图像的边缘点和噪声点等。进一步,如果只取在二维直方图中对角线附近的那些点向灰度轴z作投影来重新构成一个新的一维直方图,则该直方图应该比原直方图更加峰谷分明。反之,如果只取远离对角线上那部分点,这些点的灰度平均值也可以作为阈值用来分割图像。

② 灰度-梯度幅度法。本方法的过程原理和上述灰度-平均灰度法类似。只不过在二维直方图中选用$|\nabla f|$作为第二个轴。区域内点因为梯度幅度小,都位于横轴z,而边界点在$|\nabla f|$较大的区域里,故由梯度值较小的点构成的直方图应比原直方图有更明显的峰谷,而梯度值大的点的平均灰度值也是一个好的阈值。图7.19是本方法的一个示意图,图上的二维直方图有三个聚类中心,梯度大的聚类中心由边界点构成,若简单地把二维直方图向z轴投影所得直方图凹谷不深,则去掉$|\nabla f|$大的中心后再投影,可得到明显的双峰直方图。图中也画出了选择阈值的示意图,对于区域内点用谷点对应的阈值T_1,而对$|\nabla f|$大的点用大梯度点平均灰度值对应的阈值T_2。

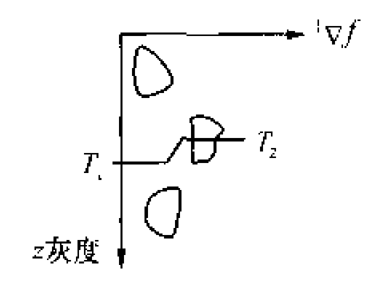


图7.19　灰度-梯度幅度法示意图

③ 基于边界对比度的阈值确定。通常认为,合理的阈值应该取在区域边界上灰度变化最剧烈的地方。因此,可以把某个阈值T所产生的边界两边灰度对比度的大小作为衡量的标准,找出能够检出最大平均边界对比度的阈值来。

对某个阈值T,先定义由边缘上的点对构成的集合$E(T)$,它包含了图像中所

有满足下列条件的空间相邻的点对。其条件是要求这些点对中的一个点的灰度值小于或者等于 T,并且另一个点的灰度值大于 T,即

$$E(T)=\{((i,j),(k,l))\mid ①\text{点}(i,j)\text{和}(k,l)\text{相邻};② \min\{f(i,j),f(k,l)\}\leqslant T<\max\{f(i,j),f(k,l)\}\} \tag{7.3-14}$$

式中:$((i,j),(k,l))$表示坐标为(i,j)和(k,l)的两个相邻的像素构成的点对。再定义由阈值 T 检出的总对比度为

$$C(T)=\sum_{((i,j),(k,l))\in E(T)}\min\{|f(i,j)-T|,|f(k,l)-T|\} \tag{7.3-15}$$

则由阈值 T 检出的边缘点的平均对比度为

$$A=C(T)/N \tag{7.3-16}$$

式中:N 是属于$E(T)$的点对的个数。于是根据平均对比度找到最佳阈值 T_b,使平均对比度为最大。用这种方法得到的阈值可以检出更多的高对比度的边界。

(4) 自适应阈值和多阈值。有些图像的直方图呈现多峰状,这时为了分割图像,可以选择多阈值进行分割。例如图像由一个背景和两个不同灰度的物体构成,这时图像的直方图呈现三个峰,这样用两个谷底处的灰度值作为两个分割阈值 T_1 和 T_2 来区分这三类。类别数越多,仅用直方图靠多阈值进行分割就越困难。

而当图像由一个物体目标和背景构成,但物体目标和背景的灰度范围有部分重叠,如果这时只取一个固定的阈值会产生较大的误差,同样也可以选择两个阈值 T_1 和 T_2,设 $T_1<T_2$,然后进行分割,即

$$f_T(m,n)=\begin{cases}1, & \text{当 } f(m,n)\leqslant T_1\\ 0, & \text{当 } f(m,n)\geqslant T_2\end{cases} \tag{7.3-17}$$

而对于 $T_1<f(m,n)<T_2$ 范围内的像素,因为两类的灰度有重叠,仅由灰度值无法判断。这时可以利用空间信息来判断,例如,用在该像素邻域内已经做出结论的点的多数来确定该点的归属。

实际情况常比上面讨论的要复杂得多,简单套用上述公式仍难以得到满意的结果。例如,当照明不均匀时,在暗光照区内白纸的灰度值可能和亮光照区里黑字的灰度值相差不多,甚至更暗。这时,不论用哪个阈值都无法兼顾亮区与暗区。在这里我们采用可变阈值来解决这个问题。在亮区阈值取得高,暗区阈值取得低,即对每个像素(或小区域)都自适应地选用不同的阈值。阈值的选取可以这样来进行:将图像分成许多小块,先对每个小块定一个阈值,各小块的阈值可以不同,然后进行对阈值适当的平滑以便消除块间阈值的突变。至于每一块阈值的确定,可以

有不同的准则和方法，一般应当先区分小块只包含一类(全部是背景点或全部是目标点)还是包含了两类。如果某小块包含了两类的像素(可以从它的直方图有双峰，其直方图方差较大等特点判断)，则可以用前面几小节所介绍的方法中的任一种来定阈值。一般来说，小块的直方图峰谷较明显，确定阈值比较容易。如果某小块只包含了某一类的像素点，其直方图较集中，呈单峰状，仅从该小块的信息难于确定阈值，要靠它四周直方图呈双峰小块的阈值，通过内插来求得该块的阈值。

(5) 基于过渡区的阈值[19]。实际的数字图像中的边界是有宽度的。一方面它将不同的区域分割开来，具有边界的特点；另一方面，它的面积不为零，具有区域的特点。该类区域称为过渡区。过渡区是一个环绕目标边界的带状区，可以通过确定过渡区保证真实边界的范围，这样可以防止分割误差偏大。

可以通过对图像有效平均梯度(Effective Average Gradient，EAG)的计算和对图像灰度的剪切操作来确定过渡区。设 $f(m,n)$ 为数字图像函数，$g(m,n)$ 表示图像 $f(m,n)$ 的梯度幅值图，则 EAG 定义为：

$$\text{EAG}=\frac{\sum_{m,n\in \mathbf{Z}} g(m,n)}{\sum_{m,n\in \mathbf{Z}} p(m,n)} \tag{7.3-18}$$

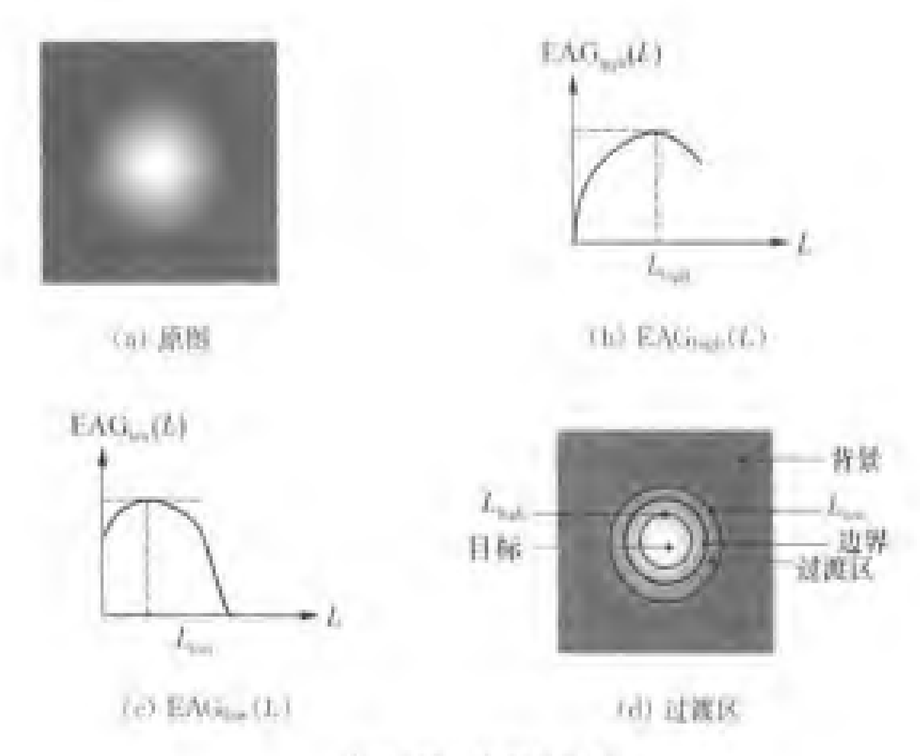
式中：$\mathbf{Z}$ 为整数集合；$p(m,n)=\begin{cases}1, & g(m,n)>0\\ 0, & g(m,n)=0\end{cases}$

由此定义可知，EAG 的计算只和非零梯度的像素有关，EAG 是图中非零梯度像素的平均梯度。为了进一步减少各种干扰，进行称之为高端剪切和低端剪切的两种操作，这两种操作利用了剪切部分的灰度值和全图灰度值的关系。设 L 为剪切值，则高端剪切和低端剪切分别为：

$$f_{\text{high}}(m,n)=\begin{cases}L, & f(m,n)\geqslant L\\ f(m,n), & f(m,n)<L\end{cases} \tag{7.3-19}$$

$$f_{\text{low}}(m,n)=\begin{cases}f(m,n), & f(m,n)>L\\ L, & f(m,n)\leqslant L\end{cases} \tag{7.3-20}$$

对剪切后的两个图像分别求梯度，则得到与剪切值 L 有关的两个 EAG(L)。这样 EAG(L)对应于高端和低端的剪切可分别写成 $\text{EAG}_{\text{high}}(L)$ 和 $\text{EAG}_{\text{low}}(L)$。典型的 $\text{EAG}_{\text{high}}(L)$ 和 $\text{EAG}_{\text{low}}(L)$ 曲线都是单峰曲线，都各有一个极值，极值点分别为 L_{high} 和 L_{low}。这两个极值点在灰度上确定了过渡区的范围，如图 7.20。



(a) 原图　(b) $EAG_{high}(L)$　(c) $EAG_{low}(L)$　(d) 过渡区

图 7.20　过渡区示例

过渡区处于目标和背景之间，而边界就处于过渡区之中，并且过渡区所包含像素的灰度值一般在目标和背景区域内部像素的灰度值之间，因此利用过渡区可以确定合适的分割阈值。例如选取过渡区内像素的平均灰度值。

7.3.2 特征空间聚类

特征空间聚类即根据某些特征将目标点(像素)划分到相应的类别中。该方法是一个较简单有效的分割方法。其基本原理是分析图像空间的元素目标的某些特征，并用相应的特征空间点表示这些元素，并根据一定的条件将特征空间的点聚集分类，然后将这些类映射回原图像空间，这样来分割图像空间。图像的特征可以是多方面的，例如灰度、颜色、纹理和信息熵等等。例如，基于灰度直方图的阈值分割可以看作一种特征空间聚类，这里以像素的灰度为特征，其直方图对应着特征空间，则选择的阈值将特征空间分为两类，然后将这两类重新映射回图像空间，则得到两种不同灰度的区域。该方法也常常采用多个特征综合考虑。一般来讲，特征空间聚成的类常对应着场景中的目标对象。典型的特征空间聚类方法有：K-均值聚类法[20]、ISODATA 法[21]、模糊 C-均值法[22]等等。下面我们主要说明 K-均值聚类算法，关于其他的聚类方法可参阅相关的文献。

K-均值聚类算法可以将一幅图像分割成 K 个区域。它是一种常用的分割算法。设(m,n)代表数字图像的像素的坐标，$f(m,n)$代表像素(m,n)的灰度。算法是要最小化如下误差函数：

$$\varepsilon^2 = \sum_{j=1}^{K} \sum_{(m,n)\in Q_j^{(i)}} |f(m,n) - \mu_j^{(i+1)}|^2 \qquad (7.3-21)$$

式中：K 表示图像中共有 K 个类(区域)，$Q_j^{(i)}$ 表示在第 i 次迭代后图像中第 j 个类(区域)，$\mu_j^{(i+1)}$ 表示 $i+1$ 次迭代后第 j 个类的平均值。式(7.3-21)实际上给出了每个像素与其对应类均值的距离和。具体的 K-均值聚类算法步骤如下：

(1) 初始化 K 个类的均值 $\mu_1^{(1)}$，$\mu_2^{(1)}$，…，$\mu_K^{(1)}$，例如可以选择每个初始类中的某个像素灰度值作为该类的初始均值。

(2) 在第 i 次迭代，输入图像的每个像素 (m, n)，并根据下述距离准则将像素归为 K 类之一。

$(m, n) \in Q_l^{(i)}$ 若 $|f(m, n) - \mu_l^{(i)}| < |f(m, n) - \mu_j^{(i)}|$

式中，$j=1, 2, \cdots, K$。该式表明将每个像素归到均值离它最近的一个类。这样所有的像素处理完成后，得到更新后的类 $Q_j^{(i+1)}$。

(3) 对每个类，按照下式更新该类的均值 $\mu_j^{(i+1)}$。

$$\mu_j^{(i+1)} = \frac{1}{N_j} \sum_{(m,n)\in Q_j^{(i)}} f(m, n)$$

式中：N_j 是更新后的类 $Q_j^{(i+1)}$ 中的像素个数。

(4) 计算误差函数 ε^2，如果 ε^2 不变(聚类成员不变)，则算法收敛结束；否则，重新从步骤(2)继续下一次迭代。

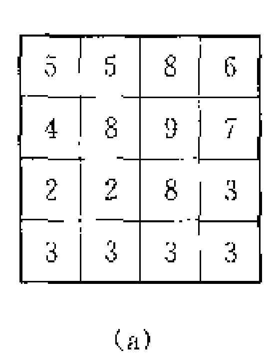
运用 K-均值聚类算法的效果受到类数目 K 的影响，可以通过先验知识得到，也可以通过试探法来确定 K。当采用试探法时，常定义一个聚类品质，根据该品质因素确定适当的类别数，常用的判别准则是根据分割后类内和类间特征值的散布图分析，即要求类内接近而类间差别要大。

7.3.3 区域生长法

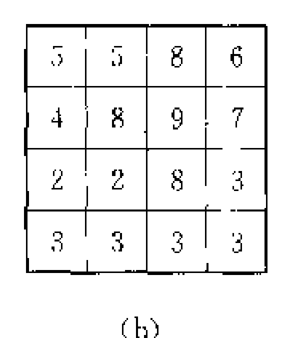
图像是由具有特殊含义的不同区域构成的，每一个区域自身具有“相似”的特性，即在一个区域内部满足某种特征的一致性。区域生长算法(Region Growing)就是基于上述特点的。其基本思想是将具有相似性质的像素(或小区域)合并起来构成区域。具体的实现过程是先对每个待分割的区域找一个生长种子(像素或小区域)作为生长的起点，根据一定的生长或相似规则，将种子周围邻域中的与种子有相同或相似性质的像素(小区域)合并到种子所在的区域中。然后将这些新像素(小区域)作为新的种子重复进行上述的生长过程，直到满足生长终止条件或者再没有满足条件的像素(小区域)可被包括进来。这样就长成了一个区域。

图 7.21 给出了一个区域生长的例子。设有一个数字图像，如图 7.21(a)所示。

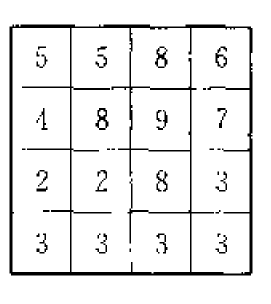
这个例子以灰度最大值点作为种子，该点的灰度值为 9。本例采用的生长相似性准则就是邻点的灰度级与已生成的区域的平均灰度级的差小于 2。第一次区域生长得到三个灰度级为 8 的邻点，如图 7.21(b)。此时这四个已接受点的平均灰度为 $(8+8+8+9)/4=8.25$，故第二次区域生长只得到灰度级为 7 的一个邻点，如图 7.21(c)所示。此时，这五个已接受点的平均灰度级为 $(8+8+8+9+7)/5=8$，因为已经没有灰度级大于 6 的邻点，生长过程终止。



(a)



(b)



(c)

图 7.21　区域生长示例

从区域生长的实现过程来看，区域生长法需要解决三个问题：①选择一组能正确代表待分区域的种子；②确定在生长过程中将相邻像素(小区域)包含进来的判定准则；③确定让生长过程停止的条件。

种子的选取直接影响到区域生长法的最终效果。其选取可以是人工介入的方式，也可以是全自动的方式。人工介入的方式实际上就是人工进行选择一组适当的种子，这种方式选择的种子一般具有代表性。全自动选择种子的方式通常要视具体问题的特点进行。当对图像有一定的先验知识时，可借助先验知识选择种子。例如，红外图像中目标的辐射比较大，也就是比较亮，所以可选用图像中最亮的像素作为种子。当图像没有先验知识时，则可借助生长所用准则对每个像素进行相应计算，如果计算结果呈现聚类的情况，则接近聚类重心的像素可取为种子。

生长准则的确定不仅依赖于具体问题的本身，还和图像数据及图像性质有关。这对于区域生长的结果是很关键的，需要综合考虑。例如是否考虑像素邻近区域灰度的均匀性和非均匀性会直接影响到分割结果。当分割彩色图像时，仅用单色的准则肯定会影响到最终分割效果。

后面将介绍几种典型的区域生长算法，它们的区别主要在于生长准则的不同。

一般生长过程停止的条件就是当生长过程进行到没有满足生长准则的像素时。当然也可以利用一些全局性的性质或先验知识来设定停止条件。

1. 单连接区域生长

单连接区域生长就是把图像中的每一个像素看成是连接图中的一个节点，然后按照一定的生长准则把特性相似的像素节点合并连接起来。图 7.21 就是一个

典型的单连接区域生长的示例。对于单连接区域生长法最简单的生长准则就是比较两相邻像素的灰度之差，如果小于设定的阈值则将它们合并。该方法虽然简单，但是效果很不理想。为了进一步提高性能，减小噪声等因素的影响，改进生长准则。下面介绍一种改进准则：设 O 为已接受的一小块区域，检查它的全部邻点，生长准则根据邻点的灰度级相对于 O 的平均灰度级接近程度来确定。另外还有另一种生长准则就是对于每个像素节点，用该节点周围 $k\times k$ 邻域的平均灰度来表示其特征，这就提高了抗干扰性。

2. 基于子区域合并的区域生长

基于子区域合并的区域生长方法比单连接区域生长更为有效。该方法不再以单像素为基础生长点，而是考虑以小区域作为生长单位。其基本步骤是：①把图像分成互不重叠的小区域；②选定一组子区域作为种子生长点；③按照生长准则比较两邻接子区的特征相似性，如果符合准则，则将新的子区并入到种子区域中；反之，将不符合准则的相邻子区视为未分割标记。④设定终止准则，反复进行步骤③将各子区域合并直到终止条件满足。

通常采用邻接子区域的灰度分布相似性作为生长准则。设 $h_1(z)$，$h_2(z)$分别为两邻接子区域的灰度直方图，则这两个直方图的相似性通常利用直方图之间的距离公式来表示，距离和相似性具有相反的属性，距离越小，相似性越大。直方图之间的距离（相似性）常用下面的这两种方法进行计算。如果两子区域的灰度直方图之间的距离小于给定的阈值，说明这两子区域的直方图很相似，这样就将这两个子区域合并。

(1) 蒙高洛夫-斯米诺夫(Kolmogorov-Smirnov)法：

$$\max_{z} |h_1(z) - h_2(z)| \tag{7.3-22}$$

(2) 平滑差(Smoothed-Difference)检测：

$$\sum_{z} |h_1(z) - h_2(z)| \tag{7.3-23}$$

基于子区域合并的区域生长算法需要注意两点：将图像分割成的子区域的尺寸大小极大地影响了分割效果，当尺寸太小时检测直方图的相似时可靠性降低，这是由于尺寸相当小时，相当于单连接区域生长，易受噪声等影响；当尺寸较大时，一些不属于同一区域的像素被可能被设定在同一个区域中，一些小的目标区域可能被忽略掉。因此需要根据实际情况确定合适的子区域大小。另外，在检测直方图相似性方面，式(7.3-23)要比式(7.3-22)效果要好，因为它考虑了所有的灰度值。

7.3.4 区域-分裂合并法

树结构可以充分描述一幅图像，树的根表示整幅图像本身，树的叶子则代表每个像元。基于这种描述，区域生长法实际上是从树的叶子开始，由下而上最终到达树的根，这样最终分割出图像的区域。与区域生长法不同，Horowitz 和 Pavlidis 则提出了一种基于新的分割思想的算法[21]，并首次应用于图像分割，该算法称之为分裂-合并分割方法(Split-merge Algorithm)。该方法从图像树的某一层开始，按照某种区域性质的一致性测度，对满足测度的相邻块进行合并，对不满足测度的块再进一步分裂为小块，这样最终得到物体目标。该方法和区域生长法一样都是以同一目标区域的性质一致性为前提的。通常分裂-合并分割方法以图像金字塔形层次结构(借助四叉树描述方法 Quadtree)作为基本的数据结构进行图像分割的。这种分割思想充分利用了图像全局和局部信息和层次性结构，从而能够实现较快的计算速度和较好的分割效果。

1. 图像数据结构描述

设数字图像 $f(m, n)$ 的尺寸为 $N \times N$，其中 $N = 2^n$。设图像的金字塔数据结构的最底层的单元包含四个像素，将四个像素组合成一个小方块，再将四个小方块组合成大方块，如此直到合成整幅图像，即得到图像的金字塔数据结构表达。因此结构是从 2×2 到 $N \times N$ 逐次递增的，结构的总层数为 $n+1$，在第 l 层 $(0 \leqslant l \leqslant n)$，方块的边长为 $s = N/2^l$。利用四叉树表达方式可将图像的金字塔数据结构形式组织起来。图 7.22 描述了如何采用四叉树具体地组织金字塔结构(这里把一些区域内性质相同的节点直接作为叶子，没有再继续分割)。其中，R 代表整个图像，R_i 代表子区域。

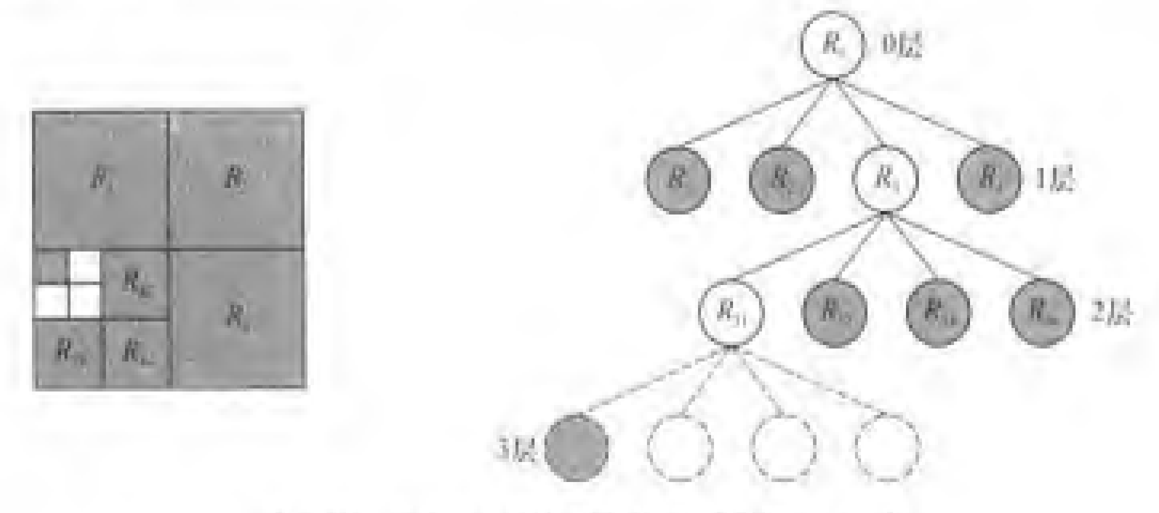


图 7.22 基于四叉树结构的分裂-合并算法示意图

Horowitz提出的分裂-合并分割方法采用数组形式来实现它的数据结构，但这种方法难以进行连接编组和小区域消除过程，这样造成内存消耗大，实现复杂且运算慢。为了克服这些问题，一些学者提出了许多的数据结构表示方法，如区域邻接图(Region adjacency graphs, RAG)[24]和重叠四叉树结构为代表的动态数据结构[25]。

2. 算法的基本原理

设R表示整幅图像区域，选择逻辑谓词P。对R进行分割的一种方法就是反复将分得的结构图像再次分裂为四个小区域R_i或合并性质相似的区域，即如果$P(R_i)=\text{FALSE}$，那么就将图像区域R_i继续分为四个区域，同时如果相邻的区域满足$P(R_j \cup R_k)=\text{TRUE}$，则这两个相邻的区域$R_j$和$R_k$进行合并。这样直到对任何区域$R_i$，都满足$P(R_i)=\text{TRUE}$。一般情况下，采用四叉树的描述方式对算法的数据结构进行组织。为了进一步说明分裂-合并分割算法，给出图7.23。图7.23是分裂-合并分割算法各步骤的一个示例。设图中阴影部分为目标，白色部分为背景，这两部分的灰度值分别都是常数。逻辑谓词P(一致性测度)是判断区域R中的所有像素的灰度值是否相同。

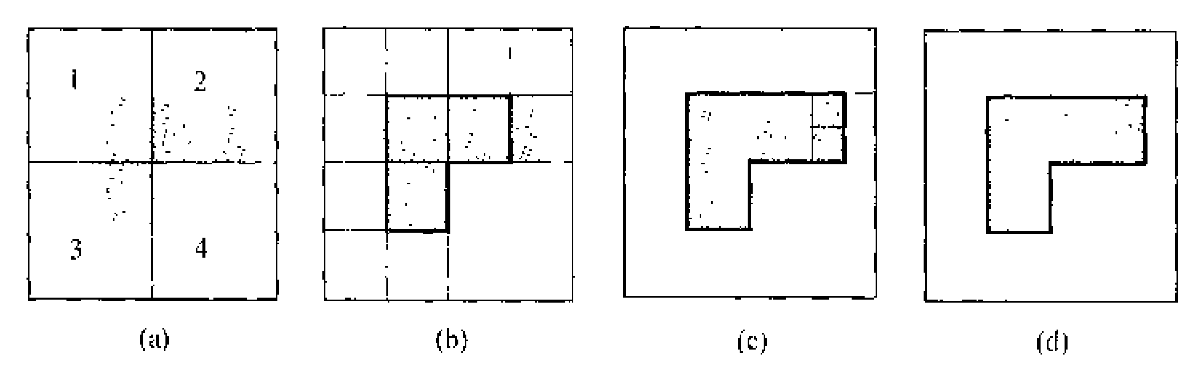


图7.23 分裂-合并分割算法各步骤示例

对于整幅图像R，$P(R)=\text{FALSE}$，将其分裂成如图7.23(a)所示的四个正方形区域。其中，右下角区域4满足P，所以不再分裂，其他三个区域不满足P，继续分裂得到图7.23(b)。这时除了目标右部一个区域不满足P外，其他的区域可以按照$P=\text{TRUE}$的准则进行区域的合并，这样得到初步的目标轮廓，如图7.23(b)中粗线所包围的区域。继续对那个$P=\text{FALSE}$的区域分裂，如图7.23(c)。然后进行最后一次合并得到整个目标区域，如图7.23(d)中粗线所包围的区域。当用四叉树方式对图像数据结构进行描述(如图7.22)。则在此情况下，理论和实验都表明，分裂-合并算法从金字塔结构的中间层开始则可以节约很多的计算时间。分裂-合并分割算法具体步骤是：

(1) 初始化。选择适当的区域一致性测度P，将图像分成金字塔形结构的子

图像，并用四叉树方式描述。从中间某层 k 开始，方块的边长为 $s = N/2^k$，求出各块的一致性测度指标。

(2) 分裂。设定允许误差阈值 e，如果某区域 R_i 不满足一致性测度，即一致性测度指标大于 e，则将节点分裂为四个小方块，并计算各小方块的一致性测度指标。

(3) 合并。进入 R_i 所对应节点的四个节点 node_{ij}，其中 $j = 1, 2, 3, 4$。如果四个子节点 node_{ij} 有公共父节点，且四个子节点的一致性测度指标小于 e，则将这四个节点区域合并成一个区域，进入到这四个节点的父节点。

(4) 组合。将非父节点或不同层的节点进行合并。当相邻区域 R_i 和 R_j 的一致性测度指标都小于 e，则将这两个区域合并，这两个区域大小可以不一样。

3. 区域的一致性测度

区域分裂合并法的一个技术关键就是如何确定区域的一致性测度。它将决定区域是否分裂或者合并。区域的一致性测度通常根据区域的统计特性进行设定，其中最常用的测度多基于灰度的统计特性。常用的一致性测度准则有：

(1) 区域中灰度最大值 M_k 与最小值 m_k 之差或方差是否小于阈值 e。其中，下标 k 表示第 k 层。若 $M_k - m_k > e$，则区域继续分裂。

(2) 两区域平均灰度之差及方差是否小于阈值。即设两个区域 R_i 和 R_j，则 R_i 和 R_j 的平均值和均方误差分别为：

$$c_i = \frac{1}{N}\sum_{(m,n)\in R_i} f(m,n) \qquad c_j = \frac{1}{N}\sum_{(m,n)\in R_j} f(m,n)$$

$$\varepsilon_i^2 = \frac{1}{N}\sum_{(m,n)\in R_i} [f(m,n) - c_i]^2 \qquad \varepsilon_j^2 = \frac{1}{N}\sum_{(m,n)\in R_j} [f(m,n) - c_j]^2$$

若 $|\varepsilon_i - \varepsilon_j| < T_0$，则两区域进行合并。若某区域的标准差 $\varepsilon > T_1$，则该区域进行分裂。

(3) 两区域的灰度分布函数之差小于阈值，通常比较两区域的直方图的相似性，可见 7.3.3 小节。

7.3.5 形态学分水岭算法

分水岭(Watershed)是地形学中的经典描述方法之一。图像处理领域中的形态学分水岭分割算法的思想就是源于地形学的这种描述。形态学分水岭分割算法具有良好的分割性能，因此一经提出，便受到重视，并得到了很好的应用。

1. 基本思想

首先对图像进行三维立体化处理，即：保持原图的两个坐标，用图像中每点的灰度作为该点的海拔高度，这样形成一幅地形图。经过上述处理，该"地形"包括山

峰和山谷间盆地，这些山峰和盆地对应于图像中不同灰度的区域。这样在地形图中存在三类点：①局部最小值的点。②当一个水珠滴在某点时，该点上的水珠只能滚到该区域的最低点。③当水珠滴在某点的位置，该点上的水珠会等概率地滚向不同区域的最低点。设存在一个局部最小值的点 a，由满足条件②的所有点构成的区域称为该点 a 的"汇水盆地"(Catchments Basin)；满足条件③的所有点构成的峰线称为"分水线"或"分水岭"。

分水岭分割算法是以图像中区域的像素近乎一致的特点为基础，其主要目标就是找出分水线，该算法的基本思想可以通过浸没模拟(Immersion Simulation)来描述：将图像看成地形表面，假设在图像的每个区域的最低处刺了一个孔，并且让水以均匀的上升速率从洞中涌出，从低到高淹没整个地形。水开始从刺过孔的海拔最低的底谷开始往上涌。当水面到达其他流域的底谷时，这些刺过孔的底谷也开始渗水。当处于不同的汇聚盆地中的水将要聚合在一起的时候，我们筑起一道假想的坝来阻止聚合。浸没过程结束时，每个流域被水淹没，并被坝完全包围。这些筑起的坝确定了对应流域的分水岭，因而对应了相应的轮廓。该模拟演示过程如图 7.24。

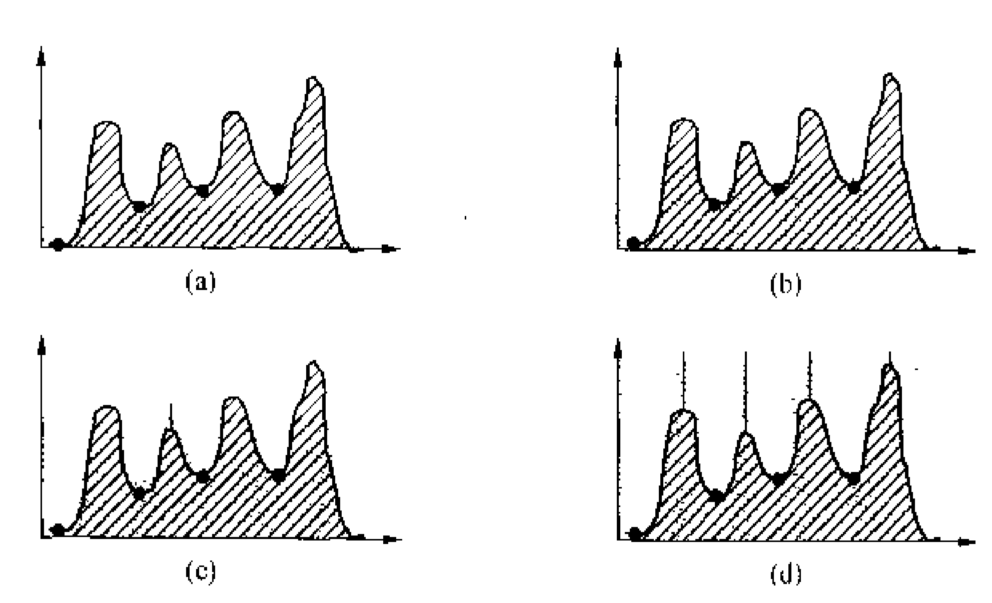


图 7.24 分水岭算法浸没模拟(一维)

假设图 7.24(a)是一个地形图的剖面，则构成了一维地形图。首先确定这个地形的所有局部极小点，如图中黑点所示，并假设在每个局部极小点处刺穿一个小孔，然后让水以均匀的上升速率从孔中涌出，从低到高淹没整个地形(见图 7.24(b))。为了防止两个不同的汇水盆地的水汇聚到一起，在它们的相接处建立起一个水坝(见图 7.24(c))，当地形被完全浸没在湖面以下之后，所有的水坝就构成了分水岭(见图 7.24(d))。

2. 水坝的构造

构造水坝的方法之一就是使用数学形态学中的形态膨胀，关于形态膨胀的具体理论参见附录2。下面我们通过一个例子说明如何通过形态膨胀的方法构造水坝。设存在两个汇水盆地 B_1 和 B_2，M_1 和 M_2 分别是这两个汇水盆地 B_1 和 B_2 的区域极小值的点，B_1^{n-1} 表示经过第 $n-1$ 次形态膨胀后汇水盆地 B_1 中被淹没的所有点的集合，B_2^{n-1} 表示经过第 $n-1$ 次形态膨胀后汇水盆地 B_2 中所有被淹没的所有点的集合。B^{n-1} 表示 B_1^{n-1} 和 B_2^{n-1} 的合集。图7.25说明了如何使用形态膨胀构造水坝。图7.25(a)两个灰色的区域分别为第 $n-1$ 次形态膨胀后的两个连通分量 B_1^{n-1} 和 B_2^{n-1}，图上的数字表示形态膨胀的次数。这两个连通分量最后融合为一个连通分量，实线框表示该连通分量。图7.25(b)为用于膨胀的结构元素，阴影部分为原点。使用图7.25(b)结构元素膨胀图7.25(a)中的两个连通分量。膨胀过程必须满足两个条件：①膨胀必须在图7.25(a)实线框所给出的连通分量的范围内进行。②膨胀过程将在引起内部的两个连通分量汇聚的点上停止。从图7.25(c)可以看出经过五次形态膨胀，这两个连通分量开始汇聚，对汇聚点（网纹方格表示）进行标记，并构建水坝，膨胀过程在汇聚点处停止。经过十次的膨胀，满足上述两个条件的点构成了只有一个像素宽度的连通路径（网纹方格表示）。这就是我们所希望得到的水坝（分水线）。水坝的高度通常设定为图像的最大灰度加上1。

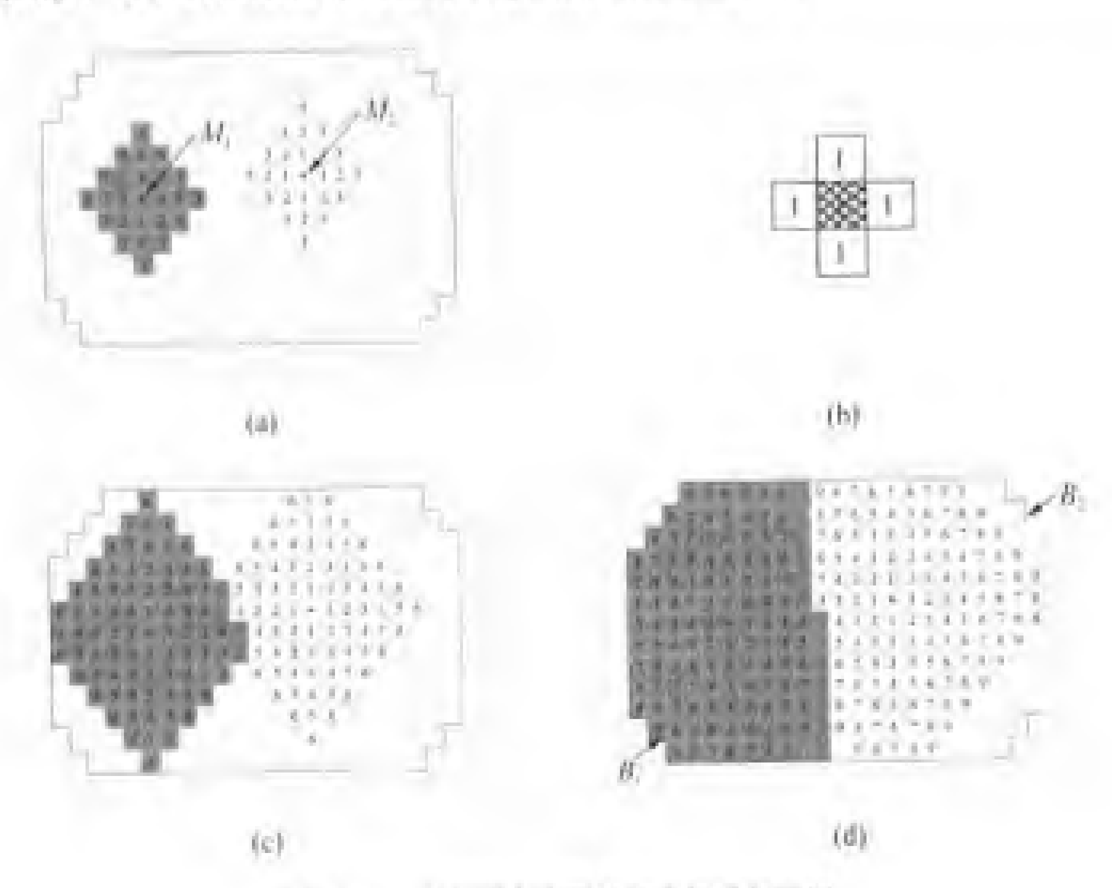


图7.25 利用形态膨胀构建水坝过程示例

3. 分割算法

设 M_1、M_2，…，M_K 分别是图像 $f(i, j)$的 K 个局部最小值点，B_1，B_2，…，B_K 分别表示 M_1、M_2，…，M_K 所对应的各汇水盆地中所有点的集合，B_k^n 表示在第 n 个阶段汇水盆地 B_k 中所有被淹没的点的集合，B^n 表示所有 B_k^n 的合集。f_{min} 和 f_{max} 代表图像 f 的最小值和最大值。根据上述定义，得到公式

$$B^n = \bigcup_{k=1}^{K} (B_k^n) \tag{7.3-24}$$

而所有的汇水盆地的合集为：

$$B^{f_{max}+1} = \bigcup_{k=1}^{K} (B_k) \tag{7.3-25}$$

$T(n)$表示图像中被水淹没的点的集合，即 $T(n) = \{(i, j) \mid f(i, j) < n\}$，则有：

$$B_k^n = B_k \cap T(n) \tag{7.3-26}$$

这样可以构造一个对应于原图的二值化图像，若原图的点 $(i, j) \in T(n)$，则该点设为 0，否则为 1。这样通过“0”的数目可以知道在第 n 阶段处于水位之下的点的数目。下面具体介绍分水岭算法的步骤。

(1) 图像预处理并提取图像的局部最小值点。由于噪声和其他如梯度的局部不规则性等因素的影响，造成图像中存在大量隐含的局部最小值。如果不进行处理，则分割所得到的最终区域会过多，这称为“过度分割”，这样的分割结果有时会毫无用处。因此为了将很小的细节对图像的影响降至最低，可以用一个平滑滤波器对图像进行平滑滤波。对处理过的图像提取局部最小值点 M_1，M_2，…，M_K。

(2) 设定起始汇水盆地的合集 $B^{f_{min}+1} = T(f_{min}+1)$。然后算法进入递归过程。将水位以整数量从 $n = f_{min}+1$ 到 $f_{max}+1$ 增加，记录每一阶段处于水位之下的点 $T(n)$。设第 n 个阶段，$T(n)$中有 Q 个连通分量 B_q^n，其中 $q \leqslant Q$。设第 $n-1$ 阶段已经构造好 B^{n-1}。对于每个 q，比较连通分量 B_q^n 和 B_{n-1}，二者关系有三种可能：

① $B_q^n \cap B^{n-1} = \varnothing$，说明出现一个新的连通分量，即汇水盆地。

② $B_q^n \cap B^{n-1}$ 包含 B^{n-1}的一个连通分量。说明 B^{n-1}存在该汇水盆地。

③ $B_q^n \cap B^{n-1}$ 包含 B^{n-1}中两个或更多的连通分量。

当满足条件①时，则将 B_q^n作为新的汇水盆地并入 B^{n-1}；当满足条件②时，则将 B_q^n 与 B^{n-1}中相应的部分合并；当满足条件③时，则需要在 B_q^n内建立分水线，用以阻止盆地间的水汇合，具体的构造方法参照本节的水坝构造部分。当所有的 B_q^n 比较完成，则构成第 n 阶段的 B^n。

(3) 后处理阶段。该阶段对分割的结果进行处理，通常按照一定的合并准则

合并一些无用的小区域，以减少过度分割。

7.4 基于边缘和区域的联合分割[27]—[30]

基于边缘或基于区域的图像分割算法各有优缺点。对于分割较为复杂的图像，仅用一种分割算法分割图像难以取得令人满意的结果。而基于边缘或者基于区域的图像分割方法具有一定的互补性，可以将这两种方法进行融合，实现优势互补，从而减少一些分割问题，提高分割的总体性能。例如，对于基于边缘的分割算法，一般因为噪声等影响会造成分割后的边缘不连续，这时需要应用边界连接算法对其进行连接处理；而基于区域的分割算法，分割的结果边缘一般是连续的，但往往分割得到的区域因为某些原因属于误分区域，这样可以将两者的优势结合，设计性能较优的联合分割算法。联合分割算法充分利用了图像的边缘和区域两方面的信息。一些专家学者对这种联合分割问题进行了深入的研究，提出了许多关于这方面的融合算法。这些分割算法的分割效果一般要优于单独使用其中一种分割算法。

根据两种类型的分割算法的融合阶段的不同，联合分割可以分为：嵌入式(Embeded)联合分割和后处理式(Post-Processing)联合分割。嵌入式联合分割就是首先对图像进行一种类型(基于边缘或者基于区域)的分割，得到的分割信息作为必要参考信息，定义另一种类型分割算法的参数或者逻辑谓词(准则)，然后用这种新的算法对图像重新进行分割。后处理式联合分割就是采用两种类型(基于边缘和基于区域)的分割算法分别对图像进行分割，这样得到两种分割信息，即边缘信息和区域信息，然后将这两种信息按照一定的融合准则进行处理，从而得到较好的分割结果。

7.4.1 嵌入式联合分割

嵌入式联合分割实际上就是根据得到的边缘信息或者区域信息，重新定义另一种类型的分割算法。这样实现了一种类型的分割算法嵌入到另一种分割算法中，从而可得到较优的分割算法完成对图像的最终分割。根据这两种分割算法的不同选择，嵌入式联合分割又可分为两类：边缘信息内嵌式和区域信息内嵌式。对于嵌入式联合分割算法，大部分类型是边缘信息嵌入式联合分割。图7.26给出了嵌入式联合分割的示意框图，其中图7.26(a)为边缘信息内嵌到区域分割算法的嵌入式联合分割；图7.26(b)为区域信息内嵌到边缘检测算法的嵌入式联合分割。

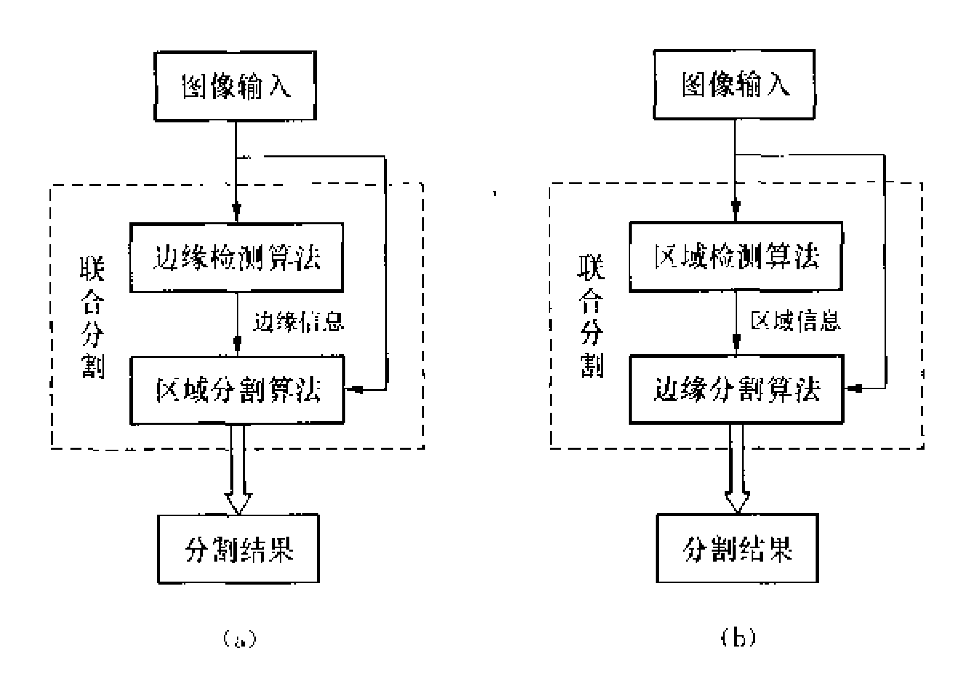


图 7.26 嵌入式联合分割算法的两种实现方案

1. 边缘信息内嵌式

对于区域分割算法，其初始状态或者决策准则是很重要的，直接影响到分割结果的优劣，一般需要一些先验知识进行设置，例如区域生长算法的初始种子(Seed)的选择问题。而对图像首先进行边缘检测，将得到的边缘信息作为先验知识，据此设置区域分割算法的初始状态或决策准则，这样可以在一定程度上避免区域分割算法缺乏先验知识的问题，改进分割效果。边缘信息主要通过两种方式作用于区域分割算法：初始种子选择和测度准则设置。

(1) 边缘信息应用于选择初始种子。对于区域分割而言，如果随机性地选择初始种子点，这会造成分割效果较差。为了得到较好的分割结果，可以将边缘信息作为先验知识，用来选择初始种子点的最优位置。例如，对于区域生长算法，区域的生长必须从区域的内部开始生长，因此种子的位置应该选在区域的内部，而不能选在边界上或者边界附近，更进一步，种子应该远离边界位置。因此我们可以根据第一步得到的边缘信息来确定种子的位置，避免所选择的种子处于边界位置上或者边界附近。

(2) 边缘信息应用于测度准则的设定。典型的区域分割算法主要有区域生长算法、分水岭算法和分裂合并算法等。这三个典型算法基本上都是基于区域一致性这个测度的。而如何定义区域的一致性这个测度是分割算法的关键问题。可以利用边缘信息结合其他的信息重新定义一致性测度，从而进一步改善分割的效果。在不同的区域分割算法中，边缘信息应用于一致性测度定义的方式大体相同，只是稍有不同。例如在灰度图像的区域生长算法中，一般通过比较当前像素灰度和区

域的平均灰度来判决该像素是否属于此区域，实际上就是判决是否满足一致性测度。但是当二者的差别很小时，则难以判决该像素是否真正属于此区域，此时可借助边缘信息进行判决。如果当前像素属于边界点，则意味着生长过程已经到达该区域的边缘，区域的生长需要在该点停止。

2. 区域信息内嵌式

区域分割得到的区域信息也可以应用到基于边缘的分割算法中。这利用了区域内部的一致性和相邻区域之间的相异性，用以判决边缘分割的结果正确与否或者定义边缘分割算法的判决准测，还可以定义初始状态。通常将区域信息应用于主动轮廓模型(Active Contour Model, ACM 或 Snake)分割的情况较多，即可以利用区域信息得到 Snake 模型的初始轮廓或者定义模型的能量函数。

Snake 算法的初始轮廓对于分割效果是很重要的，通常可由人工确定初始轮廓。为了实现自动分割，并进一步提高分割效果，可以利用区域分割得到的区域信息定义 Snake 算法的初始轮廓，即把不同区域间的不太精确的边缘作为 Snake 算法的初始轮廓。这样联合分割过程分为两步：首先对图像进行区域分割，然后对得到的区域信息进行边缘检测等处理，从而得到 Sanke 算法的初始轮廓。再是对图像进行 Snake 分割，得到更为精确的边缘。如图 7.27 所示。

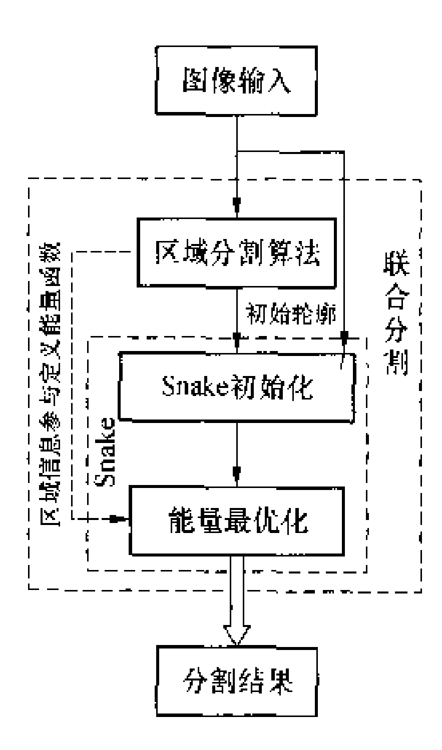


图 7.27 区域信息应用于 Snake 模型的示意框图

另外借助于区域信息，可以对 Snake 能量函数的进行重新定义。这样得到的 Snake 算法通常称为 ARM(Active Region Model)算法[26]。该算法把区域分割得到的区域信息作为全局区域模型，根据图像的局部区域信息和该全局区域模型之间的匹配情况，重新定义 Snake 能量函数的外能部分，从而影响 Snake 曲线。例如，当判断图像某局部区域处于全局区域模型的某区域的内部且具有高度一致性，说明该局部区域不包含边缘，Snake 曲线不会通过该局部区域，即此时 Snake 外能最大。图 7.27 也同时给出了 ARM 算法框架。

7.4.2 后处理式联合分割

后处理式联合分割就是首先用基于边缘和基于区域的分割算法分别对图像进行分割，得到的边缘信息和区域信息按照一定的融合准则进行融合处理，以实现精确和有意义的分割。因此后处理式联合分割具有后验式融合的特点。图 7.28 给

出了后处理式联合分割的示意框图。

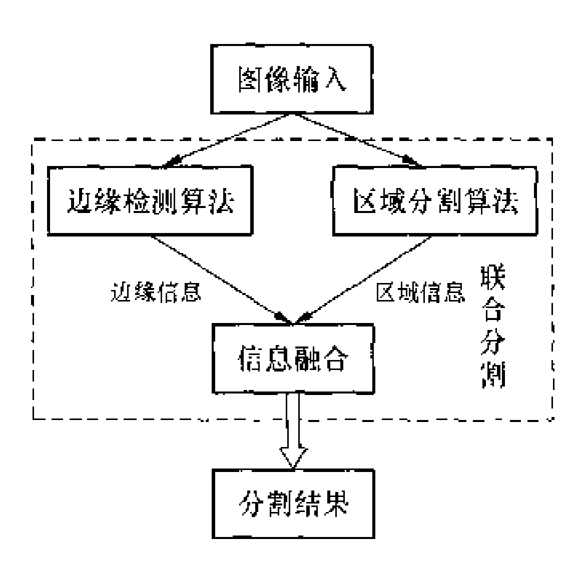


图 7.28　后处理式联合分割的示意框图

根据融合方式的不同，后处理式联合分割又可以分为三类：过分割型（Over-Segmentation）、边缘更新型（Boundary Refinement）和评价选择型（Selection-EvalUation）。

1. 过分割型

对于基于区域的分割算法，充分的一致性测度是较难定义的，在一般情况下，一致性测度定义得较为严格，例如通常区域灰度基本相同时，则认为该区域满足一致性测度，但实际上，区域的灰度可能呈线性变化。这些原因会产生很多错误的边缘，造成过分割问题。这种过分割通常是不利的，但也可以被利用。利用过分割产生较为准确的区域边缘的方法称为过分割型的联合分割。该类型的联合分割主要是通过基于边缘（基于区域）的分割算法按照一定的方式得到过分割的边缘（区域），然后利用基于区域（基于边缘）的分割得到的区域信息（边缘信息）按照一定的准则消除错误的边缘。

过分割结果可以是通过区域分割得到的区域过分割结果，也可以是通过边缘检测得到的边缘过分割结果。对于区域过分割结果，消除错误的区域边缘的方法就是对比区域分割的区域边缘和边缘检测得到的边缘是否一致，如果二者一致，则认为是正确的边缘，否则，消除这个边缘，合并边缘两侧的区域。另外可以对比较处理后得到的边缘作进一步处理，根据真实的边缘一般具有大的梯度值，如果边缘的梯度值小于给定阈值，则认为该边缘为假边缘，可以被消除。对于边缘过分割结果，通过得到的区域信息消除错误边缘。真实的边缘两侧的区域具有较大的差异性，而错误边缘两侧的区域的性质具有较大的相似性，可以进行判断该边缘是否为错误边缘。

2. 边缘更新型

有些基于区域的分割算法可以较好地检测到真实的区域，但由于噪声等因素的影响，提取的区域的边缘是很不规则的，与真实的边缘有较大的偏差。而基于边缘的分割算法可以得到较真实的边缘，因此把区域分割得到的区域作为初始区域，只是这些初始区域的边缘很不规则，可利用边缘分割得到边缘信息来进一步确定这些区域的边缘，使它们更接近于或处于真实边缘的位置。该方法与前面介绍的过分割型比较相似。

3. 评价选择型

对于分割质量的评价而言，如何定义评价准则是很关键的。若后处理式联合分割的融合准则是基于评价函数的，那么这种后处理式联合分割称为评价选择型联合分割。许多这种类型的算法是根据边缘分割得到的边缘信息定义评价函数的，通过改变区域分割算法的参数或者阈值从而得到一系列不同的分割结果，然后用评价函数选择最好的分割结果，如图 7.29 所示。

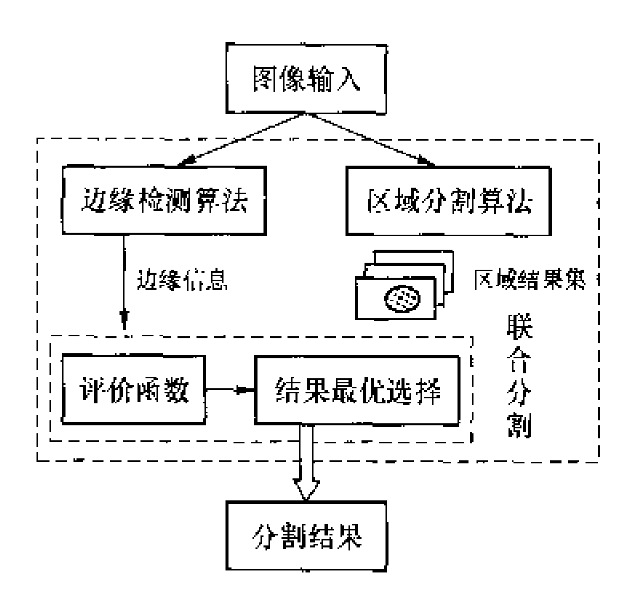


图 7.29 评价选择型后处理式联合分割框图

根据边缘信息定义的评价函数通常是根据由区域分割算法得到的区域边缘和由边缘检测算法得到的边缘之间的一致性程度来定义的。当二者完全一致时，这样的区域分割结果为最优的。这种选择策略为区域分割算法的传统难题提供了一个解决方案，例如阈值的适当选择等。

7.4.3 小结

本节对各种联合分割算法作了一个介绍，但不是很充分，更为具体的算法，可以查阅相关的文献[27]～[30]。需要指出的是，联合分割算法是图像分割算法的

一个发展方向,它实现了基于区域的分割算法和基于边缘的分割算法的优势互补,因此与单独类型的分割算法相比具有较大的优势。而对于联合分割算法的一个关键就是基于区域的分割算法和基于边缘的分割算法如何互补(即如何进行融合),虽然在这方面进行了大量的研究,但仍然有很多问题需要解决,需要进一步完善。

7.5 图像分割算法的评价[31—35]

前面几节介绍了许多图像分割方法,但还没有一种通用的分割算法能适用于所有的图像,绝大部分分割算法通常有一定的适用范围。而对于某个具体实际图像,为了得到满意的分割结果,通常会面临一个如何选择合适的分割算法的问题,特别是对分割算法不是很了解的情况下,这时就需要参考分割算法的一些性能指标来进行选择。这就涉及到分割算法的评价问题。尽管在分割算法的评价方面已有了较深入的研究,但是到目前为止还没有一个大家都能接受的通用的客观评判标准,这也正是这个问题的难点所在。对不同分割算法作一个比较评价是必需的,这是一个有意义但困难的问题。下面将介绍一些常用的评价标准。

7.5.1 评价方案和要求

分割算法的评价通过对图像分割算法性能的估测以达到分割算法优化的目的,这样可以提高现有算法的性能,也对研究新的分割技术具有指导意义。对于大多数图像处理问题而言,最后的兴趣是人,因此这时可以由人进行判断分割结果的优劣。这就是分割算法的主观评价。但不同的人对同一个问题可能有不同的评价,并且在很多情况下人是不能参与评价过程的,这样就需要对分割算法进行客观评价。图像分割算法的评价可以分成两种情况:

(1) 性能刻划(Characterization)。即同一分割算法在不同的图像情况下,分割的性能比较。

(2) 性能比较(Comparision)。比较不同算法在分割相同图像时性能的比较。

对分割评价算法的一般要求包括:评价算法的通用性、评价算法的复杂性、可靠性、是否需要参考图和是否需要人工参与等。一般认为对分割算法的客观评价可以通过分析和实验两种方式进行。因此客观评价方法可以分为分析法和实验法两大类。分析法就是直接分析分割算法本身的原理和性能;而实验法是通过对测试图像的分割结果来进行评价的。虽然分析法可以免除一些实验因素的影响,但缺乏分割效果的支持,具有较大的局限性。而通过对实际分割结果的分析来评估分割算法具有实际意义。有关实验法的评价方法已经提出较多,大致可以分为优

度法(Goodness Method)和偏差法(Discrepancy Method)。优度法一般是以建立在人的视觉基础上的某些图像特征为依据来评价分割效果;偏差法通常需要先提供一个理想分割图像作为比较标准,据此来计算实际分割效果和理想分割图像的差别,但可供比较的图像样本往往比较难被统一认定。图 7.30 给出了一个对图像分割算法进行评价的框架示意图。

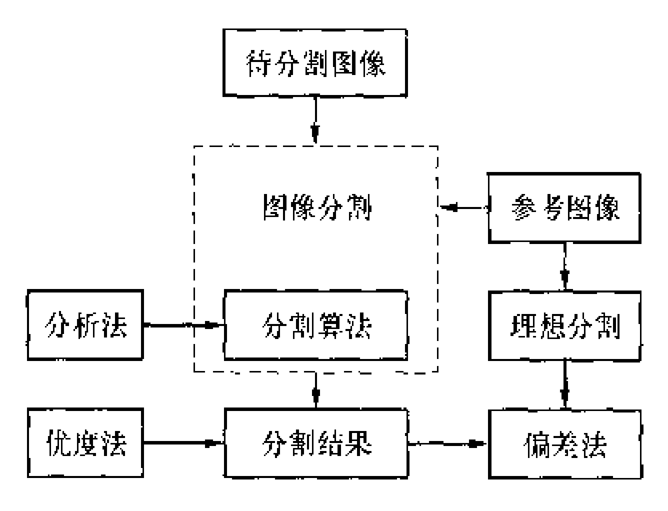


图 7.30 对图像分割算法进行评价的框架示意图

7.5.2 图像分割算法评价准则

本小节将介绍一些常用的图像分割算法评价标准。

1. 分析法

分析法直接分析分割算法本身,主要考虑算法的原理、复杂性、所需的先验知识、检测概率比和分辨率等等。算法的原理是指所采用的分析手段和理论,往往采用先进的理论方法会得到较准确的结果。而算法的实现策略为串行或者并行实现,并行实现的处理速度要比串行实现策略快很多。算法的复杂性直接影响到算法的效率和计算速度。检测概率比被定义为正确检测概率与错误检测概率之比,这个值越大表明检测的可靠性越高。早期的图像分割是仅由数据驱动的,而随着研究的深入,高层知识对分割的作用得到应用。这些先验知识的应用对分割算法的性能有较大的影响,因此可以根据分割算法所应用的先验知识的种类和数量的不同,从一定程度上比较算法的优劣。图像分割的分辨率一般可以通过分析得到,例如算法可以得到单像素或者亚像素宽度的边界。但是,并不是所有的性能都能够通过分析而得到,而且仅仅从算法本身的原理及性能出发的分析,往往缺乏足够的说服力。分析法通常为其他方法提供辅助性的信息,它很少被单独使用。

2. 实验评价方法

实验法是通过对测试图像的分割结果来进行评价的。该评价方法在实际应用中用得最多。它主要分为优度法和偏差法。下面分别介绍这两种方法。

(1) 优度法。优度法考虑了一些人的视觉特性，并以这些特性为依据来评价分割效果。该类方法在没有准确的参考分割结果的情况下，主要依据分割结果计算某种测度来评价分割的好坏，因此选择哪种测度作为评价量度是很关键的。常用的测度有区域特征一致性、区域间对比度、区域形状和基于模糊度的综合测度等等，下面主要介绍区域特征一致性和区域间对比度这两个测度，其他测度可以参阅文献[31]—[34]。

① 基于区域特征一致性的评价方法。分割得到的区域内部一般具有相似特性，因此可用各区域内部特性的一致性程度来描述分割结果。设 R_i 表示第 i 个区域，则区域内部灰度一致性测度(Uniformity)可以表示为：

$$UM = 1 - \frac{1}{\beta}\sum_i \left\{ \sum_{(m,n)\in R_i} \left[f(m,n) - \frac{1}{A_i}\sum_{(m,n)\in R_i} f(m,n) \right]^2 \right\} \tag{7.5-1}$$

式中：β 为归一化系数；A_i 为第 i 个区域的面积，即 R_i 中的像素总数。以此类似，还有熵等测度。

② 基于区域间对比度的评价方法。根据区域间的特性对比度也可以评价分割质量。对于图像中两个邻接的区域 R_i 和 R_j，设它们的平均灰度分别是 C_i 和 C_j，则它们之间的灰度对比度(Gray Level Contrast, GC)为：

$$\mathrm{GC} = \frac{|C_i - C_j|}{C_i + C_j} \tag{7.5-2}$$

注意，式中的 C_i 和 C_j 也可以是其他的特征量。

(2) 偏差法。偏差法需要首先提供一个理想的分割图像作为比较标准，依此来计算实际分割与理想分割的差别。在一般情况下，偏差法要比优度法较为有效，因为它需要一个对比测试。它主要包括基于误分像素点的测度、基于误分像素点位置的测度、基于区域数目一致性的测度等。

① 基于误分像素点的测度。由于分割错误而产生误分的像素个数是一个重要的图像分割质量评价指标。设 $M(k)$ 表示第 k 个区域的误分系数，即

$$M(k) = \frac{\sum_{i=1}^{N} C_{ik} - C_{kk}}{\sum_{i=1}^{N} C_{ik}} \tag{7.5-3}$$

式中：C_{ij} 表示第 i 个区域像素划分到第 j 个区域的个数；N 为分割结果之子区域

数。再对 $M(k)$ 求均值，该均值越小，则图像的分割质量越好。

② 基于误分像素点位置的测度。实际的分割结果往往是不理想的，总会有一些像素被错误的划分。这些被误分的像素与它们本应该属于的正确区域之间的距离也可以反映分割质量的优劣。设 N 为所有被错分的像素点的总数，$d(i)$ 表示第 i 个错分像素点到其所属正确区域的最近距离[41]，这时可用归一化距离测度（Normalized Distance Measure，NDM）表示分割质量的优劣：

$$\mathrm{NDM}=\frac{\sqrt{\sum_{i=1}^{N}d^2(i)}}{N} \tag{7.5-4}$$

上式表示所有误分像素点到其所属正确区域的平均误分距离。

③ 基于区域数目一致性的测度。设 N' 为对一幅图像进行分割得到的区域个数，N 为图像中正确的区域个数，二者可能不同。它们之间的差异在一定程度上可以对分割算法进行评价。下面给出一种区域数目一致性的测度 F，该测度被定义为

$$F=\frac{1}{1+p\left|\frac{1}{N'}-\frac{1}{N}\right|^{q}} \tag{7.5-5}$$

式中：p 和 q 均是尺度参数。

7.5.3 评价方法小结

前面介绍的各个评价方法各有特点和不足，每个评价准则只能反映出分割算法性能的一个方面，在许多情况下，需要综合考虑，而不能只用一个准则评价算法的各个特性。另外，虽然现在已提出很多的评价算法，但是仍然缺乏能够全面评价分割算法性能的方法。因此如何设计能全面真实地反映分割算法性能的评价方法仍然是一个重要的研究课题。这对我们选择和改进分割算法具有重要的意义。

习 题

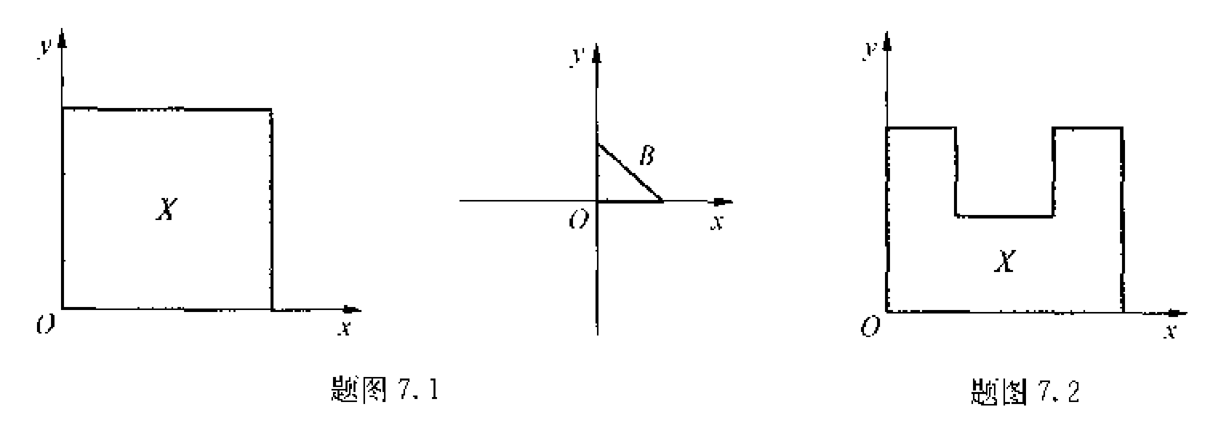
1. 试用图例说明 Sobel 梯度模板不具有各向同性的性质。
2. 试用 Hough 变换检测过原点的圆，请给出具体的算法和实现程序。
3. 试提出两种对油墨不均匀的指纹图像进行二值化的方案。
4. 设一幅图像由物体和背景构成，物体占整个图像区域的百分比为 20%，物体的

平均灰度和方差分别为100和400,而背景的平均灰度和方差分别为40和1 225,请计算将物体从背景中分割出来的最优阈值。

5. 用分水岭算法分割图像得到的分割结果存在"过分割"现象,设计两种算法对该"过分割"结果进行处理,从而得到较好的分割结果。

6. 请实现 K 均值聚类算法和区域生长分割算法程序,比较这两种算法的性能,说明初始点对分割结果的影响,说明原因。

7. 对题图 7.1 所示的 X 和 B、作图表示 $X\oplus B$、$X\ominus B$、$X\circ B$、$X\bullet B$ 的形状。



题图 7.1 题图 7.2

8. 写出用区域分裂合并算法对题图 7.2 进行分割的全过程。

参考文献

[1] MPEG 4 visual fixed draft international standard. ISO/IEC 14496-2[S]. Oct. 1998.

[2] MPEG-7 Overview (version 3.0), ISO/IEC JTC1/SC29/WG11 N3445 [S]. Geneva, 2000.

[3] Cheng K S, et al. The application of competitive Hopfield neural network to medical image segmentation[J]. IEEE-MI, 1996,15:560~567.

[4] Hall L O, et al. A comparison of neutral network and fussy clustering techniques in segmenting magnetic resonance images of the brain[J]. IEEE NN, 1992,(5):672~681.

[5] Bhanu B, Sungkee Lee, Ming J. Adaptive image segmentation using a genetic algorithm [J]. IEEE Transactions on Systems. 1995,12(25):1543~1567.

[6] D Marr and E C Hildreth. Theory of edge detection. Proc. Royal Soc[J]. London, Ser. B Vol 207, 1980,187~217.

[7] J Prewitt. Object enhancement and extraction. in "Picture Processing and Psychopictories", (B Lipkin and A Rosenfeld eds.), New York, Academic Press, 1970[J]. 75~149.

[8] J. F. Canny. "A computational approach to edge detection"[J]. IEEE Trans. Pattern Analysis and Machine Intelligence, 1986,8:679~698.

[9] 普拉特 W K. 数字图像处理学[M]. 高容坤,等译. 北京:科学出版社,1981.

[10] Kass M, Witkin A, Terzopoulos D. Snake: active contour models[J]. Proc. of First Inter Conf. on Computer Vision, London, 1987: 259~268.

[11] Cohen L D. On active contour models and balloons. CVGIP: Image Understanding[J]. 1991, 53 (2): 211~218.

[12] Xu C, Prince J L. Snakes, shapes and gradient vector flow[J]. IEEE Transactions on Image Processing, 1998, 7 (3): 359~369.

[13] Serra J. Image Analysis and Mathematical Morphology[M]. Academic Press, 1982.

[14] A. Rosenfeld and A. C. Kak, Digital Picture Processing [M]. Vol. 2, Academic Press, 1982.

[15] Rosenfeld A, Thurston M. Edge and curve detection for visual scene analysis[J]. IEEE Trans Computer, 1971, 20: 512~519.

[16] Marr D. Theory of edge detection[J]. Proc Roy Soc, 1980, B(27): 187~217.

[17] Witkin A P. Scale-Space Filtering[J]. Proc of IJCAI, 1983, 1019~1021.

[18] Mallat S, Zhong S. Characterization of signals from multiscale edges[J]. IEEE-PAMI, 1992, 14(7): 710~732.

[19] Zhang Y J, Gerbrands J J. Transition region determination based thresholding[J]. Pattern Recognition Letters, 1991, 12: 13~23.

[20] Tekalp A M. Digital Video Processing[M]. Prentice Hall, 1995.

[21] Dunn J C. A Fuzzy Relative of the ISODATA Process and Its Use in Detecting Compact Well-Separated Clusters[J]. Journal of Cybernetics, 1974, 3(3): 32~57.

[22] Bezdek J C. Pattern Recognition with Fuzzy Objective Function Algorithms[M]. New York: Plenum Press, 1981.

[23] Horowitz S, Pavlidis T. Picture segmentation by a tree traversal algorithm[J]. JACM 1976, 23(2): 368~388.

[24] Chen P C, Pavlidis T. Segmentation by texture using a co-occurrence matrix and a split-and-merge algorithm[J]. Comput Graphics Image Process, 1979, 10(2): 172~182.

[25] Burt P J, Hong T J, Rosenfeld A. Segmentation and estimation of image region properties through cooperative hierarchical computation[J]. IEEE Trans on Syst Man Cybern, 1981, 11: 802~809.

[26] Alexander, D., Buxton, B. Implementational improvements for active region models [M]. British Machine Vision Conference. Colchester, UK, 1997.

[27] Ron Schoenmakers. Integrated Methodology for Segmentation of Large Optical Satellite Images in Land Applications of Remote Sensing[M]. Luxembourg, Italy, 1995.

[28] Pavlidis T and Liow Y T. Integrating Region Growing and Edge Detection[J]. IEEE Trans. Pattern Analysis and Machine Intelligence, 1990, 12(3): 225~233.

[29] Kara Falah R, Bolon P. A region region and region edge cooperative approach of image segmentation. Image Processing, 1994. Proceedings. ICIP-94[J]. IEEE International Conference, Volume 3, 13-16 Nov. 1994 Page(s): 470-474 vol. 3.

[30] X. Munoz, Freixenet J. Strategics for image segmentation combining region and boundary Information[J]. Pattern Recognition Letters, v 24, n 1~3. January, 2003,375~392.

[31] Zhang Y J. A Survey on Evaluation methods for Image Segmentation[J]. Pattern Recognition, 1996,29(8):1335~1346.

[32] Zhang Yujin. A review of recent evaluation methods for image segmentation. Signal Processing and its Applications, Sixth International, Symposium on[J]. 2001 Volume 1, 13~16 Aug. 2001 Page(s):148~151 vol. 1.

[33] Huang. Q. Quantitative methods of evaluating image segmentation[J]. Image Processing, 1995. Proceedings, International Conference on Volume 3, 23~26 Oct. 1995 Page(s): 53~56 vol. 3.

[34] Strasters K, Gerbrands J J. Thress-dimensional image segmentation using a split, merge and group approach[J]. Pattern Recognition Letters, 1991,12:307~325.

[35] 章毓晋. 图像分割[M]. 北京:科学出版社.2001.

[36] Milan Sonka, Vaclav Hlavac,等. 图像处理、分析与机器视觉(第二版)[M]. 艾海舟.等译, 北京:人民邮电出版社,2003.

第 8 章　图像描述

上一章讨论的图像分割技术把一幅图像分成一些具有不同特征的部分。为了对图像作进一步的分析、识别等自动处理，需要用适当的数据或符号来表示这些部分，这就是图像描述的任务。图像描述总是以被描述的对象的某些特征为基础的，因此各种描述方法总是反映了或隐含了被描述对象的某些特征。描述分为对图像中各部分的描述和对各部分之间关系的描述。本章主要讨论前者，包括线条的描述、区域的描述以及几何特征的测量。

8.1　离散图像的几个几何概念

为了便于理解图像描述的具体方法，有必要了解离散图像的几个几何方面的概念，例如图像的连通性、边界和距离等。它们有些是显而易见的，有的则要繁复的证明[1]。这些证明不在本章讨论。

一幅数字图像可以看做像素点的集合。分割后的图像一般由几个部分组成，每个部分具有一定的灰度值，若全图所有部分只含有两个灰度值，则为二值图。设图像由 m 个非空子集 $S_1, \cdots, S_m$ 组成，图像可表示为 $\bigcup_{i=1}^{m} S_i$。且 $S_i \cap S_j = \varnothing$，$i \neq j$。当 $m = 2$ 时，全图由 S 和 S^c 组成，为二值图。为方便起见，主要讨论二值图像的情况，并约定 S 中每点的值为 1，S^c 为 S 的补集，其中每点的值为 0。

8.1.1　邻接与连通

邻接和连通是像素之间的基本关系，是研究图像描述的基础。邻接和连通的概念可以直观地叙述如下。除了图像的边缘点以外，所有像素都有 8 个邻点，如图 8.1 所示。

图像处理技术中常采用两种邻接方式：一种是 4 邻接，即指水平和垂直方向的四个邻点 $(i-1, j)$，$(i, j-1)$，$(i+1, j)$，$(i, j+1)$，由它们组成的邻域称 (i, j) 的 4 邻域；第二种为 8 邻接，(i, j) 与包括对角元素在内的所有 8 个邻点为 8 邻接，相应的邻域称为 8 邻域。若两个像素 P 和 Q 是 4 邻接的，则称它们为 4 连通；如果

它们是8邻接的,则称为8连通。由于连通的定义不同,同一图像的连通性含有差别。例如,图8.2(a)是一简单的2值图,如果按4连通定义来理解,则1所表示的部分是四根不连通的线段;若按8连通来理解,则是一个闭合的环。下面再对邻接和连通作进一步的讨论。

$i-1$, $j-1$	$i-1$, j	$i-1$, $j+1$
i, $j-1$	i, j	i, $j+1$
$i-1$, $j-1$	$i+1$, j	$i+1$, $j+1$

图8.1 邻域和邻点

```
0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0
0 0 0 1 1 1 0 0
0 0 1 0 0 0 1 0
0 0 1 0 0 0 1 0
0 0 1 0 0 0 1 0
0 0 0 1 1 1 0 0
0 0 0 0 0 0 0 0
```

(a)

```
0 0 0 0 0 0 0 0
0 1 1 1 1 0 1 0
0 1 0 0 1 0 1 0
0 1 0 0 1 0 1 0
0 1 1 1 1 0 1 0
0 1 1 1 1 0 0 0
0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0
```

(b)

图8.2 连通性解释

我们说图像中两点P,Q之间存在长度为n的通路,意思是指存在一系列点$P=P_0$, P_1, …, $P_n=Q$,其中P_i是P_{i-1}的邻点,$1\leqslant i\leqslant n$。如果这些邻点的定义是4邻接的,则称4通路,如果是8邻接的,则称8通路。

定义:设S是图像中的一个子集,P,Q是S中的点。如果从P到Q存在一个全部点都在S中的通路,则称P、Q在S中是连通的。若这个通路是4通路,则称4连通;若是8通路,则称8连通。

对于S中任意一点P,S中所有的与P连通的点的集合称为S的连通分量(Component),即一个连通的区域。

连通性具有如下性质:

(1) P与P是连通的;

(2) 若P与Q连通,则Q与P也连通;

(3) 若P与Q连通,Q与R连通,则P与R连通。

由此可知,图像中的两个点,当且仅当它们属于同一连通分量时,它们才是相互连通的。

设S'是S的补集,且图像的边缘点包含在S'中,我们称包含图像边缘的S'的连通分量为背景。而S'的其他连通分量,如果存在的话,则一定处于S的某个连通分量之中,称之为孔。例如,图8.2(b)中S_1为背景,S_2为孔。S中有孔的连通分量称复连通,没有孔的称为单连通。

通常对S和S'应采用不同的连通性定义。现用图8.2(a)说明它的必要性。设1所表示的为S,0为S'。若对S采用8连通,则它是一个闭合的环,环中间的0形成了一个孔,它与环外的S'是不连通的。如果对S'也用8连通,则S'的环外部分和环内部分是连通的,都是背景,这就与环是闭合的这一点产生了矛盾。因此,

S^c 应该用 4 连通定义，反之亦然。

S 中与 S^c 邻接的点的集合 S' 称为 S 的边界，集合中的点称为边界点。S 中除去 S' 的点，即 $S-S'$ 称为 S 的内部。

8.1.2 距离

距离是像素之间重要的几何特征。所谓距离必须满足下列三条性质，设图像中两点 P, Q 之间的距离为 $d(P, Q)$，
则有：

(1) 非负性：$d(P, Q) \geqslant 0$；

(2) 对称性：$d(P, Q) = d(Q, P)$； (8.1-1)

(3) 三角不等式：$d(P, R) \leqslant d(P, Q) + d(Q, R)$。

常用的距离有三种。设 P、Q 的坐标分别为 (m_1, n_1) 和 (m_2, n_2)，则

(1) 欧几里得距离：

$$d_e(P, Q) = \sqrt{(m_1 - m_2)^2 + (n_1 - n_2)^2} \tag{8.1-2}$$

(2) 街区距离：

$$d_4(P, Q) = |m_1 - m_2| + |n_1 - n_2| \tag{8.1-3}$$

(3) 棋盘距离：

$$d_8(P, Q) = \max(|m_1 - m_2|, |n_1 - n_2|) \tag{8.1-4}$$

以上三种距离是 minkowsky 距离的特例。这种距离可以用下式表示

$$d(P, Q) = [|m_1 - m_2|^p + |n_1 - n_2|^p]^{1/p}$$

当 $p = 2$ 时，就是欧几里得距离。当 $p = 1$ 时，则成了街区距离 d_4。当 $p \to \infty$ 时，则成了棋盘距离 d_8。街区距离和棋盘距离是欧几里得距离的两种近似。由于在数字图像的处理和分析中，距离常常是作为某种特征出现的，关心的不是它的绝对精度。而这两种近似计算方便，结果又都是正整数，因此一般都采用这两种近似的度量而不采用欧氏距离。

		2		
	2	1	2	
2	1	0	1	2
	2	1	2	
		2		

(a) $d_4 \leqslant 2$

2	2	2	2	2
2	1	1	1	2
2	1	0	1	2
2	1	1	1	2
2	2	2	2	2

(b) $d_8 \leqslant 2$

图 8.3 两种距离

显然，与 P 点的街区距离 $\leqslant t$ 的点组成的区域是菱形，而与 P 的棋盘距离 $\leqslant t$ 的点组成的区域是直立的正方形。如图8.3所示，其中 $t=2$。而 $d_4(P,Q)$ 为 P 到 Q 的最短的4通路的长度。$d_8(P,Q)$ 为 P 到 Q 的最短的8通路的长度(设任意方向两邻点的距离为1)。

点 P 到一个点集 S 的距离定义为 P 到 S 中所有点的距离中的最短距离，或说 P 点到 S 中最近点的距离。点集 S 的直径定义为 S 中任意两点间的最大距离。

8.2 收缩、膨胀和细化

分割后的图像表现为背景以及分离出来的物体，在用适当方式描述这些物体以前，往往要作一些形态上的处理。上一章对形态处理已作了详细论述，这里将从另一个角度介绍某些处理方法。

8.2.1 收缩和膨胀

用比较形象的说法，收缩和膨胀就是对图像中某些区域(线和点是区域的特例)进行运算。收缩是使区域从四周同时向内缩小；而膨胀则反之，它使被膨胀的区域同时向四周扩大。因此，一般地说，若用 S 表示组成某个区域的像素点的集合，则收缩就是反复地消除区域 S 的边界点，即将 S 的边界点用 S^c 的值来代替；而膨胀则是反复地将 S 的补集 S^c 的边界点加入 S。设 S' 是 S 的边界，$(S^c)'$ 是 S^c 的边界，用 $S^{(-i)}$ 表示对 S 进行收缩，用 $S^{(i)}$ 表示对 S 进行膨胀处理，i 是膨胀和收缩的次数。那么，对 S 作一系列收缩处理可表示为 $S^{(0)}=S$，$S^{(-1)}=S-S'$，$S^{(-2)}=S^{(-1)}-S^{(-1)'}$ 等等。而对 S 作一系列膨胀处理可表示为 $S^{(1)}=S\cup(S^c)'$，$S^{(2)}=S^{(1)}\cup((S^{(1)})^c)'$ 等。显然，对任意的 k 有 $S^{(k)}=((S^c)^{(-k)})^c$，以及 $S^{(-k)}=((S^c)^{(k)})^c$。

因为收缩和膨胀都是针对某区域的边界点而言的。根据前节对边界点的定义，膨胀和收缩的处理和结果与区域的邻接定义是4邻接还是8邻接有关。另外，要指出的是膨胀收缩过程是不可互换的，也是不可逆的，就是说，区域 S 先膨胀 m 次，再收缩 n 次，不一定等于先收缩 n 次，再膨胀 m 次，即 $(S^{(m)})^{(-n)}$ 不一定等于 $(S^{(-n)})^{(m)}$，且它们都不等于 $S^{(m-n)}$。$(S^{(m)})^{(-m)}$，或 $(S^{(-m)})^{(m)}$ 也不一定等于 S。现举一简单例子来说明，设 S 仅由一点 P 组成，若先膨胀一次再收缩一次，则 $(S^{(1)})^{(-1)}=S=\{P\}$，但若先收缩一次再膨胀一次，则因收缩一次后一点消失，所以 $(S^{(-1)})^{(1)}=\varnothing$ ($\varnothing$ 为空集)。但是可以证明，$(S^{(-n)})^{(m)}\subseteq S^{(m-n)}\subseteq(S^{(m)})^{(-n)}$，当 $m=n$ 时，有 $(S^{(-k)})^{(k)}\subseteq S\subseteq(S^{(k)})^{(-k)}$

以上讨论说明，利用收缩膨胀算法可以消除图像中某些孤立的小区域或噪声点。这对分割后的进一步处理是很有用的。

对多值图的膨胀和收缩，可以利用局部极大极小的方法进行，这里不作进一步说明，请详阅参考文献[2]。

8.2.2 细化

在图像处理中，形状信息是十分重要的。为了便于描述和抽取特征，对那些细长的区域常用它的"类似骨架"的细线来表示，这些细线处于图形的中轴附近，而且从视觉上来说仍然保持原来的形状，如图 8.4 所示。这种处理就是所谓的细化。细化的目的是要得到与原来区域形状近似的由简单的弧和曲线组成的图形。前面用到"类似骨架"这一词汇是因为在后面讨论中轴变换时将较为严格地使用"骨架"这一名词。从这个意义上说，细化结果与后面提到的"骨架"不是一回事。

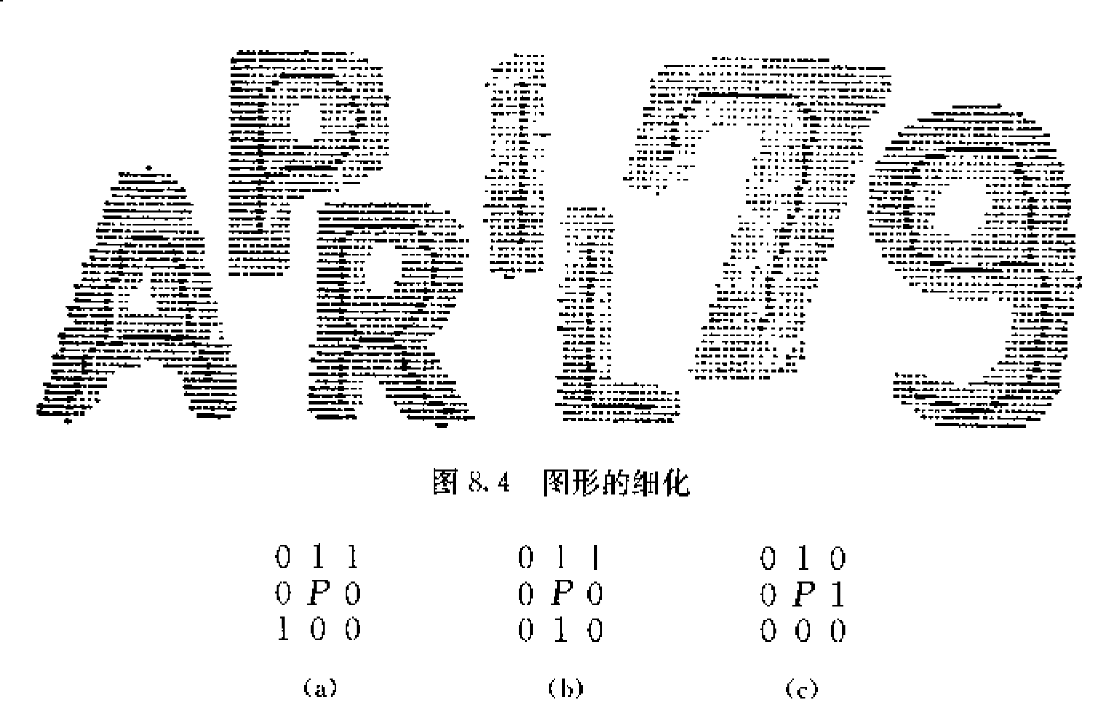


图 8.4 图形的细化

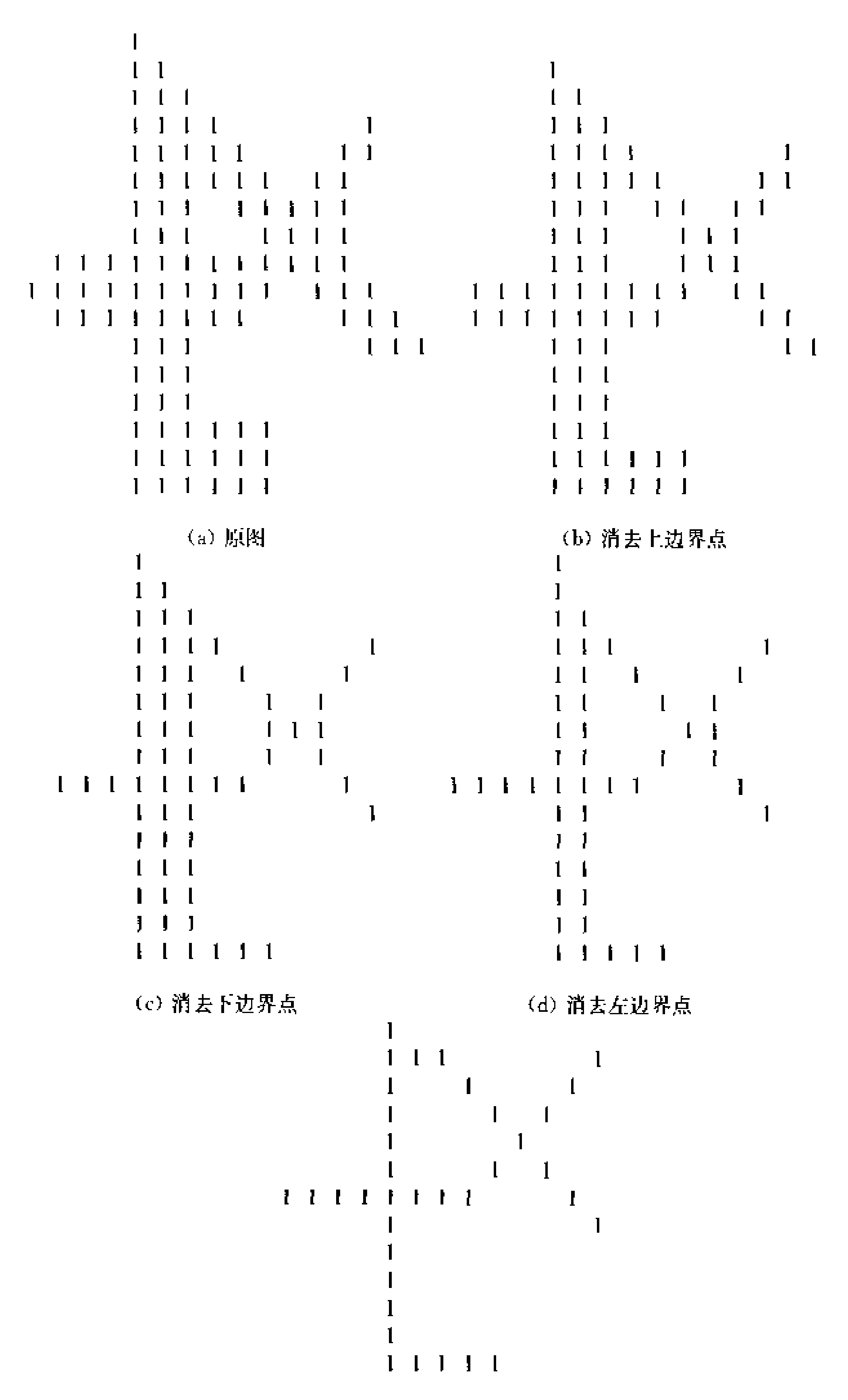
$$\begin{matrix} 0 & 1 & 1 \\ 0 & P & 0 \\ 1 & 0 & 0 \end{matrix} \qquad \begin{matrix} 0 & 1 & 1 \\ 0 & P & 0 \\ 0 & 1 & 0 \end{matrix} \qquad \begin{matrix} 0 & 1 & 0 \\ 0 & P & 1 \\ 0 & 0 & 0 \end{matrix}$$

(a) (b) (c)

图 8.5 简单边界点

弧和曲线可以这样定义：它们是集合 S 的子集 T，这个子集是 S 的连通分量，子集中除两个端点以外的每一个点都有而且只有两个邻点（端点仅有一个邻点）。

细化算法实际上是一种特殊的多次迭代的收缩算法。但是，细化的结果是要求得到一个曲线组成的连通的图形，这是细化和收缩的根本差别。所以不能像收缩处理那样简单地消去所有的连界点，否则将破坏图形的连通性，甚至完全消去图形。因此，在每次迭代中必须消去 S 的边界点而不破坏它的连通性，而且不能消去那些只有一个邻点的边界点，以防止弧的端点被消去。



(a) 原图

(b) 消去上边界点

(c) 消去下边界点

(d) 消去左边界点

(e) 消去右边界点,得结果(S为8连通)

图8.6 细化举例

首先，将 S 的边界分为上、下左、右四个方向，每次迭代只消去其中一个方向的边界点。为保证细化结果呈"骨架"状，即处于 S 的中线附近，应该交替处理相反方向的边界，即以上、下、左、右、上，…，的次序反复进行。

另外，对可以消去的边界点要给以严格的限制，为此定义"简单的边界点"。

定义：P 为 S 中的一个边界点，如果它的 8 邻域中属于 S 的点只有一个与 P 相邻接的连通分量，则 P 点为 S 的简单边界点。

定义中"与 P 相邻接"只是对 S 为 4 邻接时而言，若 S 为 8 邻接，则不必考虑这一条件，因为这时 8 邻域中任一点都与 P 邻接；根据这一定义，可以确定哪一个边界点是"简单的"。显然，对于 S 的不同的连通定义，结论是不同的。例如图 8.5，根据定义，如果 S 的定义是 4 连通的，则(a)中的 P 点是简单边界点，(b)(c)则不是。而如果 S 的含义是 8 连通的，则(c)中的 P 是简单的边界点，而(a)、(b)则不是。若去除不是简单边界点的 P 点，则将破坏 S 或 S^C 的连通性，这是不允许的。而去掉简单的边界点不会产生这样的问题。我们约定，孤立点不属于简单的边界点，而端点显然属于简单边界点。

至此，可以将细化算法归纳如下：消去 S 中那些不是端点的简单边界点，并按 S 的上面、下面、左面、右面的顺序反复进行，直至不存在可以消去的简单边界点。

为了实现以上算法，边界点的检查和消除应该"并行"地进行，而不能以串行方式逐行地进行。就是说，在一次迭代中，在对所有边界点进行检测以前，不应执行"消去"操作。例如图 8.6(a)，如果一行一行地执行"消去朝上的边界点"，则图形将从上向下一层一层被剥去，结果显然是错误的。但是这种并行处理需要用并行处理计算机。如果采用普通计算机，则可用下法进行。一种方法是分两次扫描完成一次迭代，第一次扫描检查所有边界点，并作出标记，然后在第二次扫描中消去经过标记的点；第二种方法是一行一行地扫描全图，某一行的简单边界点必须在下一行的边界点被检查并标记以后才被消去。

图 8.6 是以上算法的处理过程和结果举例，其中 S 定义为 8 连通。对细化的结果有几点要加以说明。

(1) 如果细化结果是由两个或两个以上的弧在一处相交而成，那么在接头处将会有多个点聚集在一起，形成一团。如图 8.7(a)所示。

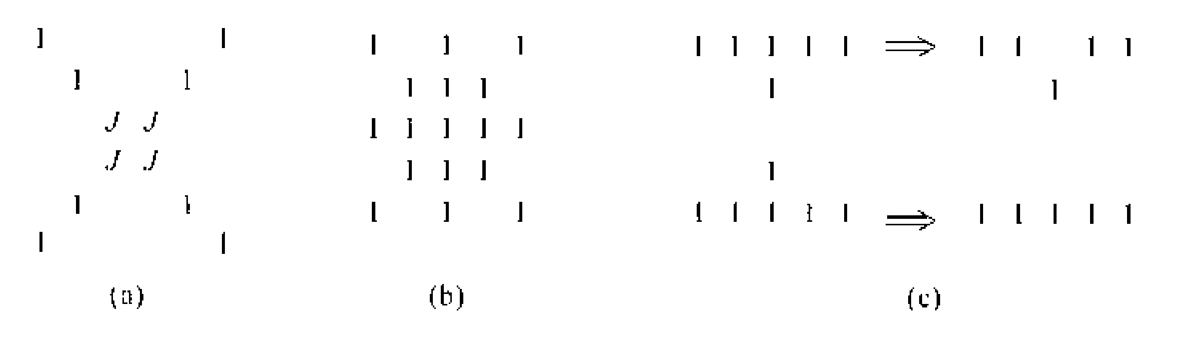


图 8.7 细化结果的讨论

(2) 有时细化结果并不完全是由单个像素宽度的弧或曲线组成，而可能存在内部的点，如图 8.7(b)所示。

(3) 同样形状的 S，当它的方向不同时，细化的结果可能不同，如图 8.7(c)所示。

以上讨论的是二值图的细化算法，这在图像分析中有广泛的应用。在某些情况下需要对多值图(灰度图)直接作细化处理。下面介绍一种由 C. R. Dyer 和 A. Rosenfeld 提出的多值图细化算法，先定义两个概念：

(1) 通路 P_1，P_2，…，P_n的强度定义为这一通路上任意点的最小值。

(2) P 和 Q 之间连通性的阶数定义为 P 到 Q 的所有通路中的最大强度。

有了这两个概念，便可以定义多值图情况的简单点。

定义：对一个点 P，若用它的邻域中的最小值来代替它而不降低 P 点的 8 邻域中任意两点连通性的阶数，则称 P 点为简单点。

多值图的细化算法可以描述如下：如果某个点 P 既是简单点又有两个或两个以上的邻点的值比它的值大，则消去这点。而且与二值图的细化类似，每次只处理一个方向，且按上、下、左、右、上，……，的次序反复进行。这里指的某个方向是指在这个方向上有一个较低值的邻点。

8.3　线的描述

数字图像中的线可能是直线、折线或者是曲线。它们可能是闭合的，也可能是不闭合的，图像中区域的边界往往是一种闭合的曲线，用一些简单的形式来描述这些线条的特征是十分有用的。本节介绍一些典型的描述方法。

8.3.1　曲线拟合

曲线拟合是一种曲线的近似表示方法，常用的有多项式拟合和 B 样条函数拟合。这类描述都以一定的误差要求为准则，最后用相应的拟合曲线的参数来简洁地描述被拟合的曲线。

1. 多项式拟合

对任意曲线都可以将其适当分段，然后用多项式进行拟合。

取曲线上一些点，形成点集 (x_i, y_i)，$i = 1, 2, \cdots, N$。假设这些点是顺序排列的(或称这些点是有因果关系的)，即(x_i, y_i)和(x_{i-1}, y_{i-1})是曲线上相邻的点，且 $x_{i-1} > x_i$。

现用一个多项式来拟合这一曲线。令多项式为 $\hat{y} = P(x)$。若用普通多项式，则

$$P(x) = a_0 + a_1 x + a_2 x^2 + \cdots + a_m x^m = \sum_{j=0}^{m} a_j x^j \tag{8.3-1}$$

式中 m 为多项式次数，且 $m \ll N$，a_j 为第 j 次项的系数。为使多项式 $P(x)$ 与点集有最佳拟合，要求拟合的误差最小。就是说，将样点集合中样点的 x_i 代入式(8.3-1)中，会得到一个相应的$\hat{y}_i$，若曲线与多项式在这一点完全拟合，则应有 $\hat{y}_i = y_i$。将所有样点的 x_i 代入式(8.3-1)，可得一组共 N 个$\hat{y}_i$ 值($i=1, 2, \cdots, N$)。拟合的目的就是找出多项式的系数 a_0，a_1，$\cdots a_m$，使 y_i和$\hat{y}_i$ 的总的误差最小。通常采用最小平方误差，即要使下式误差

$$\varepsilon = \sum_{i=1}^{N} [y_i - P(x_i)]^2 \tag{8.3-2}$$

达到最小。为此，令

$$\partial\varepsilon / \partial a_j = 0, \ j = 0, 1, \cdots, m \tag{8.3-3}$$

由此可得一线性方程组

$$\sum_{j=0}^{m} a_j S_{k+j} = T_k, \ k = 0, 1, \cdots, m \tag{8.3-4}$$

式中：

$$S_j = \sum_{i=1}^{N} x_i^j, \ j = 0, 1, 2, \cdots, 2m \tag{8.3-5}$$

$$T_j = \sum_{i=1}^{N} x_i^j y_i, \ j = 0, 1, 2, \cdots, 2m \tag{8.3-6}$$

解式(8.3-4)方程组即可得拟合多项式的系数 a_0，a_1，$\cdots$，a_m。这时，曲线就可以用这些系数简洁地表示。

由于用普通多项式拟合要解线性方程组，在某些情况下可能产生病态方程组，而给求解带来困难。若用正交多项式来拟合，则可避免产生这样的问题。

如果 $\phi_1(x)$，$\phi_2(x)$，$\cdots$，$\phi_m(x)$是区间$[a, b]$上的正交函数族，则它们必然满足

$$\int_a^b \phi_i(x)\phi_j(x)dx = 0, \ i \neq j \tag{8.3-7}$$

$$\int_a^b \phi_i(x)\phi_j(x)dx \neq 0, \ i = j \tag{8.3-8}$$

在离散情况下可写作

$$\sum_{k=1}^{N} \phi_i(x_k)\phi_j(x_k) = 0, \ i \neq j \tag{8.3-9}$$

$$\sum_{k=1}^{N}\phi_i(x_k)\phi_j(x_k) \neq 0,\ i = j \tag{8.3-10}$$

这些函数若是多项式的形式，则为正交多项式。

若将式(8.3-1)中 x^k 用 $x^{(k)}$ 来代替，即令

$$P_{mN}(x) = 1 + b_1x^{(1)} + b_2x^{(2)} + \cdots + b_mx^{(m)},\ m < N \tag{8.3-11}$$

式中：

$$x^{(k)} = x(x-1)(x-2)\cdots(x-k+1) \tag{8.3-12}$$

m 为阶数，N 为样点数。为满足正交条件，式(8.3-11)应改写成如下形式

$$P_{mN}(x) = \sum_{k=0}^{m}(-1)^k\binom{m}{k}\binom{m+k}{k}\frac{x^{(k)}}{N^{(k)}} \tag{8.3-13}$$

式中：$N^{(k)} = N(N-1)(N-2)\cdots(N-k+1)$，$m < N$。可以证明$\{P_{mN}(x)\}$是等距正交多项式。（等距指样点的 x_i 等距排列。）

为进行曲线拟合，令

$$F_m(x_i) = \sum_{j=0}^{m}a_jP_{jN}(x_i),\ i = 1, 2, \cdots, N \tag{8.3-14}$$

与式(8.3-2)相似，可以建立误差函数

$$\varepsilon = \sum_{i=1}^{N}[F_m(x_i) - y_i]^2 \tag{8.3-15}$$

为求误差最小时的系数 a_j，用与前面相同的方法，可得

$$\sum_{j=0}^{m}a_jQ_{jk} = Q_k,\ k = 0, 1, \cdots, m \tag{8.3-16}$$

式中：

$$Q_{jk} = \sum_{i=1}^{N}P_{jN}(x_i)\cdot P_{kN}(x_i),\ j, k = 0, 1, \cdots, m \tag{8.3-17}$$

$$Q_k = \sum_{i=1}^{N}P_{kN}(x_i)y_i,\ k = 0, 1, \cdots, m \tag{8.3-18}$$

由于 P_{mN} 是正交多项式，所以

$$Q_{jk} = \begin{cases} 0, & j \neq k \\ Q_{jj}, & j = k \end{cases} \tag{8.3-19}$$

因此这一方程组的系数矩阵是对角阵，因而不必求解方程组，只须用下式就可直接求得 a_j：

$$a_j = Q_j / Q_{jj}, \ j = 0, 1, \cdots, m \tag{8.3-20}$$

可见用正交多项式拟合算法简单，且避免了因方程组病态而带来的求解困难，应用效果较好[3]。

多项式次数可以这样来选择：先取一较小的值，例如取 $m=2$，利用上述方法求得 $\{a_j\}$，然后将样点的 x_i 代入多项式，可得拟合的值 $\{\hat{y}_i\}$。算出它与原曲线之间的误差，若最大误差超过允许的精度，则将多项式次数加 1，重新计算 $\{a_j\}$，直到满足要求的精度为止。

2. B 样条拟合

样条函数拟合是一种分段多项式的拟合方法。分段多项式的次数一般不超过 3 次，因为 3 次多项式既有低次多项式的简便，又比较灵活。这种拟合方法可以保证分段结合处的平滑性，用它可以描述形状比较复杂的曲线，是用得很多的曲线拟合方法。但是，一般的样条函数有一个严重的缺点，整个曲线的形状都受所有样点的影响。就是说只要有一个样点的位置发生变化，就会改变整个曲线的形状，因此很难作局部修正。另外，还存在某些数值逼近问题上的不稳定性。所以，在许多实际应用中常用一种特殊的 B 样条函数。这里，我们只介绍 B 样条函数。

B 样条是样条函数的一种。它的特点是仅在 $m+1$ 个区间内函数值不为零。m 为样条函数的次数。常用的是三次 B 样条函数，它的基函数为 $B_i(x)$，是一段非负的三次曲线。对 B 样条函数来说，区间 $x_{i+1}-x_i$ 可以相等也可以不相等。在等区间的情况下，对所有区间基函数相同。这时，三次 B 样条函数仅在 $i-2$ 至 $i+2$ 区间内不为零，如图 8.8 所示。

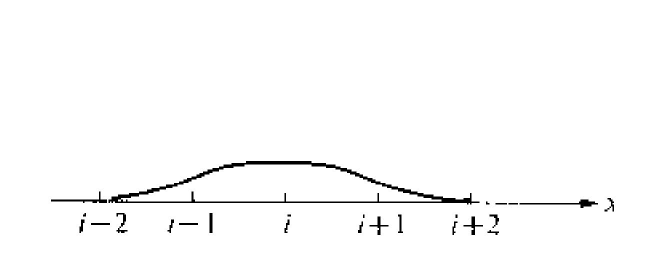


图 8.8 均匀区间 B 样条函数 $B_i(x)$

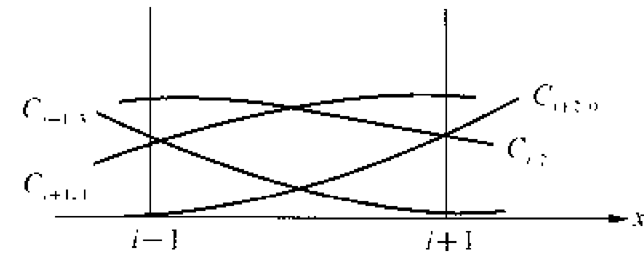


图 8.9 一个区间内样条函数的组成

用这样的基函数可以形成所谓样条函数来拟合任意曲线。设已知样点数为 N，即有点集 $\{x_i\}$，$i=1, 2, \cdots, N$，则有

$$P(x) = \sum_{i=0}^{N-1} a_i B_i(x) \tag{8.3-21}$$

为方便起见，设在样点处 x 取值为正整数，即 $x=1, 2, \cdots, N$。式中 a_i 为系数。系数共有 $N+2$ 个，其中 N 个系数由 N 个样点确定，另外两个则由边界条件决定。例如曲线端点曲率若为 0，则有 $a_1=\frac{a_0+a_2}{2}$，$a_N=\frac{a_{N-1}+a_{N+1}}{2}$。从图 8.9 可以看出，每个基函数可按区间分成 4 段，分别用 $C_{i,0}(x)$、$C_{i,1}(x)$、$C_{i,2}(x)$、$C_{i,3}(x)$ 来表示。因此对样条函数而言，在每个区间应由 4 段函数组成，即 $C_{i-1,3}(x)$，$C_{i,2}(x)$，$C_{i+1,1}(x)$，$C_{i+2,0}(x)$，如图 8.9 所示。即有

$$P_i(x)=a_{i-1}C_{i-1,3}(x)+a_iC_{i,2}(x)+a_{i+1}C_{i+1,1}(x)+a_{i+2}C_{i+2,0}(x) \tag{8.3-22}$$

因为在等间隔情况下，基函数相同，仅需在 x 轴上作平移。因此可以定义 4 段基函数为

$$c_{i,j}(x)=c_j(x-i) \quad i=0, \cdots, N+1;\ j=0, 1, 2, 3 \tag{8.3-23}$$

则在区间 $(i, i+1)$ 内，式(8.3-22)可以写成

$$P_i(x)=a_{i-1}C_3(x-i)+a_iC_2(x-i)+a_{i+1}C_1(x-i)+a_{i+2}C_0(x-i) \tag{8.3-24}$$

式中 $C_j(t)$ 表示为：

$$\begin{aligned}
C_0(t)&=t^3/6\\
C_1(t)&=(-3t^3+3t^2+3t+1)/6\\
C_2(t)&=(3t^3-6t^2+4)/6\\
C_3(t)&=(-t^3+3t^2-3t+1)/6
\end{aligned} \tag{8.3-25}$$

式中：t 为参数，$t=x-i$，在区间 $[0, 1]$ 内变化。式(8.3-24)还可用矩阵形式表示

$$[\boldsymbol{P}_i(t)]=[t^3 \quad t^2 \quad t \quad 1][\boldsymbol{C}][a_{i-1} \quad a_i \quad a_{i+1} \quad a_{i+2}]^T \tag{8.3-26}$$

式中：

$$[\boldsymbol{C}]=\frac{1}{6}\begin{bmatrix}-1 & 3 & -3 & 1\\ 3 & -6 & 3 & 0\\ -3 & 0 & 3 & 0\\ 1 & 4 & 1 & 0\end{bmatrix} \tag{8.3-27}$$

用类似方法可以求得 B 样条函数的系数 $\{a_i\}$，$i=0, 1, \cdots, N+1$。

由于由 B 样条基函数形成的函数 $P(x)$ 在每个子区间上最多由 $m+1$ 个 B 样条函数的和来确定，而 B 样条基函数本身仅在 $m+1$ 个区间不为零。这说明它确实有

局部性质,即改变式(8.3-21)中任何系数,只改变 $m+1$ 个区间中曲线的形状。

B 样条函数有连续的一阶导数和二阶导数,因此由它生成的样条曲线十分平滑。

3. 迭代的端点拟合

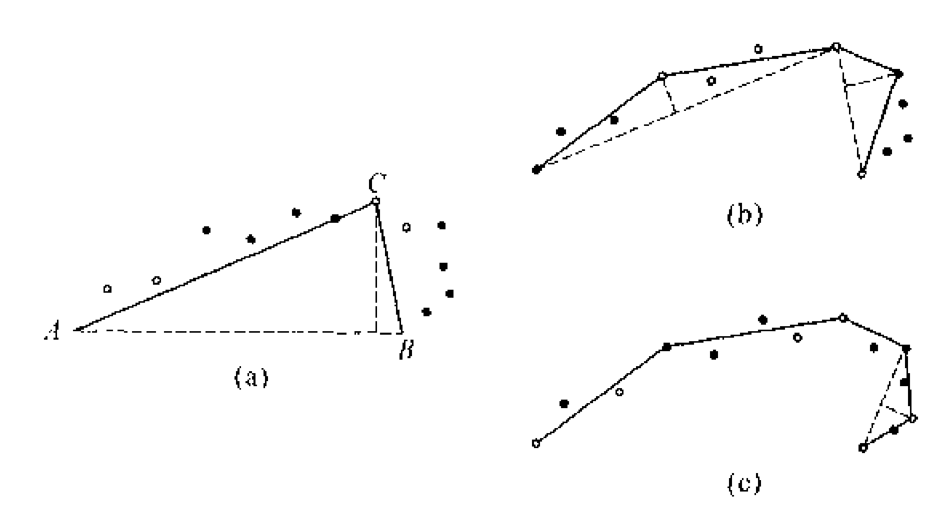


图 8.10 迭代端点拟合法

这种拟合用迭代算法来完成。设有一条由 N 个点组成的曲线,如图 8.10 所示。迭代方法可以叙述如下:第一步联接两个端点 AB。然后计算点集中所有点到连线 AB 的距离。如果所有距离都小于设定的误差门限,则拟合过程结束,AB 线段可以近似地表示这个点集。否则,找出与 AB 距离最大的点,设为 C。第二次迭代则联接 AC 和 BC 段并舍去 AB 线段,这时原 AB 分为 AC 和 BC 两段。然后对每一段重复以上过程,直到全部满足误差条件为止。这时,数据点和折线段可以很好地拟合。图 8.10 表示了这一拟合过程。

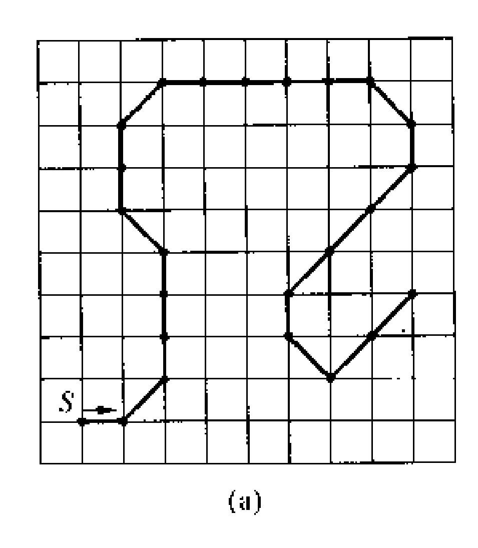
这一方法的优点是简单,缺点是不正确的点或噪声点会引起较大的误差,而且在每条线段中间的点拟合精度较差。

8.3.2 链码

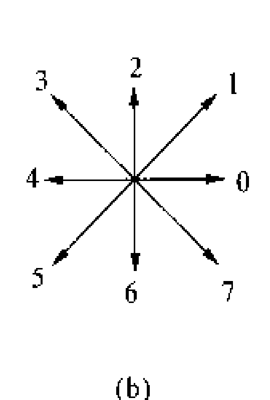
在离散图像中,线条或边界是由一串离散的像素点组成的(见 8.1.1 和 8.2.2 小节)。若用一网格覆盖图像,并使像素点位于网格的交点上,则离散图像的线条可以看做由短的线段组成的链。这些短的线段正好是网格相邻交点的连线,如图 8.11(a)所示。在这样的基础上就可以用链码来描述任意曲线或者闭合的边界。

链码又称 Freeman 链码,根据不同的连通性定义,有 4 链码和 8 链码之分。这里以 8 链码为例。链码实质上是一串指向符的序列。指向符共有 8 个方向,分别用 0,1,2,3,4,5,6,7 等八个数来表示 0°,45°,90°,135°,180°,225°和 315°等八个方向。指向符如图 8.11(b) 所示。

链码表示就是从起点开始沿曲线观察每一线段的走向并用相应的指向符来表



(a)



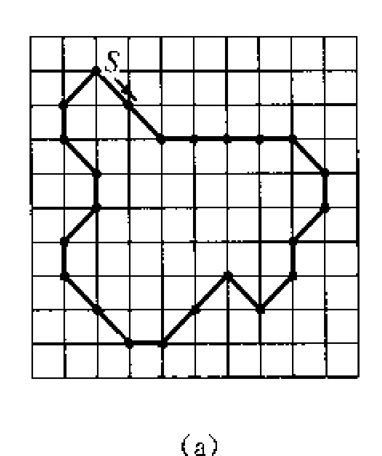
(b)

图 8.11 线条的离散表示及指向符

示，结果形成一个数列。对于图 8.11(a)所示曲线，形成的链码为：

0 1 2 2 2 3 2 2 1 0 0 0 0 0 7 6 5 5 5 6 7 1 1

若图像的采样晶格是正方形的，正方形的边长为 d，则链码中偶数码代表水平和垂直方向，长度都为 d，其余则为对角方向，长度均为$\sqrt{2}d$。这一点在几何特性的测量中是有用的。



(a)

(b) 链码 77000076565355433212321

(c) 规格化 00007656535543321232177

(d) 导数 77000076565355433212321

70100077717620770771177

图 8.12 链码的规格化和链码的导数

为了确定链码所表示的曲线在图像中的位置，并能由链码准确地重建曲线，则需要标出起点的坐标。但在有些场合，当用链码来描述闭合边界时，往往不关心起点的具体位置。由于边界是闭合的，起点与终点重合，且起点位置的变化只引起链码的循环位移。此时，常将链码规格化，即改变起点位置使链码所表示的整数最小，这通过码的循环位移很容易得到，如图 8.12(c)所示。这种规格化的表示对形状匹配是很有帮助的。

在某些场合，还采用链码的导数来描述曲线。对 8 链码来说，它的导数是指对

每个码元作后向差分,并对结果作模8运算,如图8.12(d)所示。这种表示的优点在于与边界的旋转无关(第一个码元除外,该码元是链码中原来的值,未取其导数),而且同样描述了各链的走向。确定各段走向的规则是:在当前走向的基础上旋转一个角度就是下一线段的走向,这个角度等于下一个码元数字乘以$\frac{\pi}{4}$。

利用链码除了可以表示线条和边界以外,还可以用来计算一些几何特征的量,例如长度、闭合曲线的周长、所围的面积等等。这些将在8.5节中讨论。

8.3.3 傅里叶描述子

上一小节讨论了用链码可以描述闭合的边界。这一小节介绍描述闭合边界的另一种方法——傅里叶描述子[18]。这种方法用一系列傅里叶系数来表示闭合曲线的形状特征。但是它仅适用于单封闭曲线,如图8.13(a)所示,而不能描述如图8.13(b)那样的复合闭合曲线。

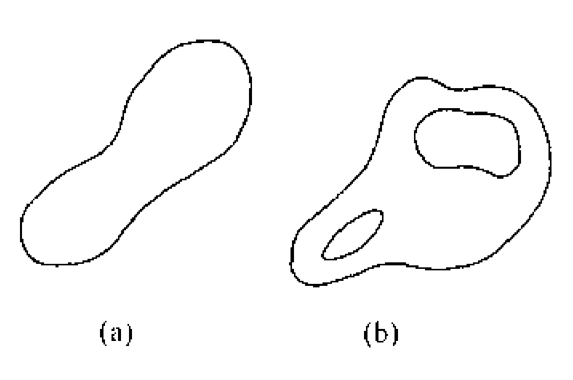


图8.13 单闭合曲线和复合闭合曲线

1. 连续曲线的傅里叶描述子

有一单闭合曲线,若从它上面一点开始按一定方向沿曲线移动,则可通过曲线上每一点回到原来的位置,并可反复循环。如果把曲线放在复平面上,则曲线可表示成

$$z(l)=x(l)+\mathrm{j}y(l) \tag{8.3-28}$$

显然这是一个以周长T为周期的周期函数,可以用傅里叶级数展开

$$z(l)=\sum_{n=-\infty}^{\infty}a_n\exp[\mathrm{j}2\pi nl/T] \tag{8.3-29}$$

式中:

$$a_n=\frac{1}{T}\int_0^T z(l)\exp[-\mathrm{j}2\pi nl/T]\mathrm{d}l \tag{8.3-30}$$

l为参量,表示从起点开始沿逆时针方向的曲线弧长。这样,就可以用一系列系数$\{a_n\}$来表示$z(l)$。但是这样的表示与起点的位置、曲线的尺寸及旋转等有关,为此作如下的修正。

若我们用带有参量l的向量表示这一曲线,则

$$z(l)=(x(l),\ y(l)) \tag{8.3-31}$$

$x(l)$和$y(l)$是曲线上长度为l这一点在x方向和y方向上的分量。设此封闭曲线

的总长为 L，而且曲线是平滑的，即它的一阶导数

$$\frac{\mathrm{d}x(l)}{\mathrm{d}l} 和 \frac{\mathrm{d}y(l)}{\mathrm{d}l}$$

存在。若在曲线长为 l 的这一点作曲线的切线，则切线与 x 轴的夹角为(图 8.14)

$$\theta(l) = \arctan\left[\frac{\mathrm{d}y(l)/\mathrm{d}l}{\mathrm{d}x(l)/\mathrm{d}l}\right] \quad (8.3-32)$$

为了消除起点的影响，可以定义一个规格化后的夹角函数 $\varphi(l)$。

$$\varphi(l) = \theta(l) - \theta(0) \quad (8.3-33)$$

式中：$\theta(0)$ 是起点处的夹角。下面再通过例子说明 $\varphi(l)$ 的性质。

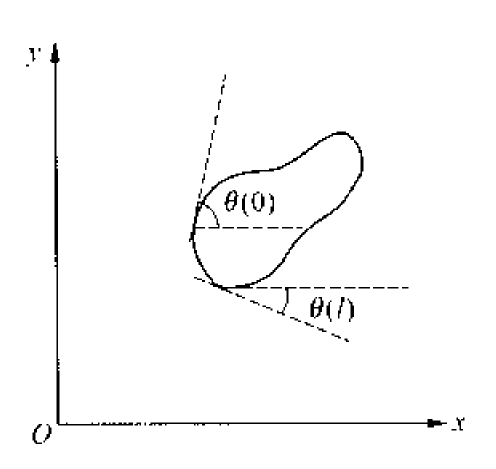


图 8.14 闭合曲线和夹角函数

例 1 有一长方形的闭合曲线，如图 8.15(a)所示。现以 S 为起点沿逆时针测量曲线的夹角函数，并以 $\varphi(l)$ 为纵坐标，l 为横坐标作 $\varphi(l)$ 的变化曲线，如图 8.15(b)。显然，$\varphi(l)$ 的变化与长方形的大小有关。因此，如果要使对形状的描述与闭合曲线的大小无关，就要进行尺度的变换。令

$$l = Lt/2\pi \quad (8.3-34)$$

这样，无论周长 L 有多大，当 l 从 0 到 L 变化时，t 总是从 0 变到 2π。而当从起点 S 起绕曲线一周后回到 S 点时，$\varphi(l)$ 也从 0 变化到 2π，即 $\varphi(L) = 2\pi$。为了形成以 2π 为周期的周期函数，必须在移动一周后减去 2π，为此令

$$\varphi^*(t) = \varphi(Lt/2\pi) - t \quad (8.3-35)$$

显然，这是一个以 2π 为周期的周期函数，如图 8.15(c)所示。

例 2 有一个半径为 r 的圆，其周长 $L = 2\pi r$。根据上面的讨论，有

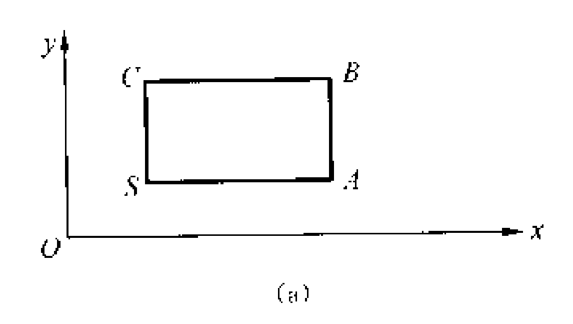
$$\varphi(l) = l/r$$

$$\varphi^*(t) = \varphi(Lt/2\pi) - t = \frac{1}{r} \cdot \frac{Lt}{2\pi} - t = 0 \quad (8.3-36)$$

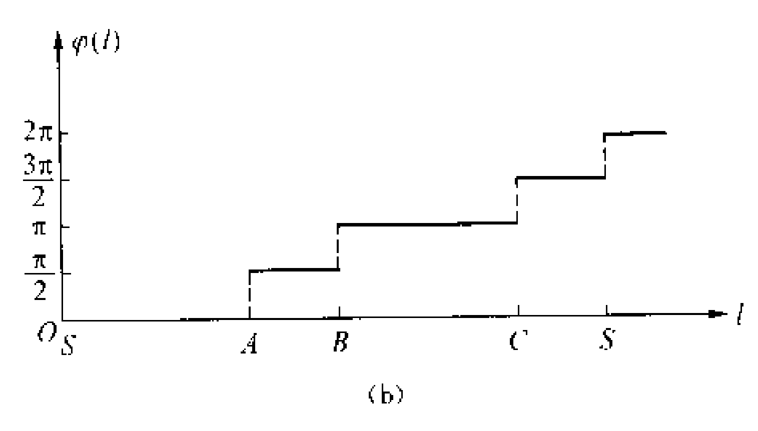
这里 $\varphi^*(t)$ 是常数，它是周期函数的一个特例。

通过这两个例子，可以看出 $\varphi^*(t)$ 是一个周期函数，因此可以将其展开为傅里叶级数，即

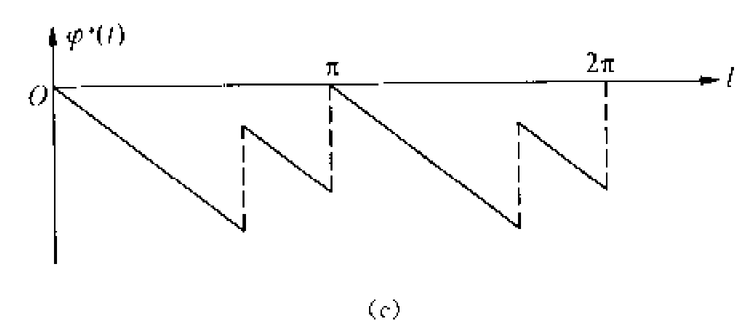
$$\varphi^*(t) = \mu_0 + \sum_{k=1}^{\infty}(a_k \cos kt + b_k \sin kt) \quad (8.3-37)$$



(a)



(b)



(c)

图 8.15 例 1 图示

或者

$$\varphi^{*}(t) = \mu_0 + \sum_{k=1}^{\infty} A_k \cos(kt - \alpha_k) \tag{8.3-38}$$

式中：

$$a_k = \frac{1}{\pi}\int_0^{2\pi} \varphi^{*}(t)\cos kt\,\mathrm{d}t \tag{8.3-39}$$

$$b_k = \frac{1}{\pi}\int_0^{2\pi} \varphi^{*}(t)\sin kt\,\mathrm{d}t \tag{8.3-40}$$

$$\mu_0 = \frac{1}{\pi}\int_0^{2\pi} \varphi^{*}(t)\,\mathrm{d}t \tag{8.3-41}$$

$$A_k = (a_k^2 + b_k^2)^{1/2} \tag{8.3-42}$$

$$\alpha_k = \arctan(b_k/a_k) \tag{8.3-43}$$

式中：μ_0为平均分量，它与闭合曲线的形状无关，因此可以用(a_k, b_k)或者(A_k, α_k)来描述这个闭合曲线。A_k和α_k是a_k和b_k的极坐标形式，A_k表示第k次谐波的幅度，α_k是第k次谐波的相角，这里$k=1, 2, \cdots, \infty$。但是在一般情况下，由于傅里叶级数收敛迅速，当k较大时，a_k、b_k或A_k均接近于零，因此常用有限的M项，即$k=1, 2, \cdots, M$时的(a_k, b_k)或(A_k, α_k)来描述这个曲线，并称之为傅里叶描述子。严格地说，项数减少后，曲线不再是闭合的了。

傅里叶级数还可以表示成复数形式，即

$$\varphi^*(t) = \sum_{k=-\infty}^{\infty} C_k e^{jkt} \tag{8.3-44}$$

式中系数C_k为复数，即

$$C_k = \frac{1}{2\pi}\int_0^{2\pi} \varphi^*(t) e^{-jkt} dt \tag{8.3-45}$$

式中：$k=0, \pm 1, \pm 2, \cdots$。$C_k$与三角形式级数的系数有如下关系

$$\begin{aligned} C_0 &= \mu_0 \\ 2C_k &= a_k - jb_k = A_k e^{-j\alpha_k} \\ 2C_{-k} &= a_k + jb_k = A_k e^{j\alpha_k} \end{aligned} \tag{8.3-46}$$

上面讨论的傅里叶描述子是以闭合曲线上的切线与x轴的夹角为基础来描述曲线的。傅里叶描述子还可以从闭合曲线上各点的曲率变化来描述这一曲线。

定义曲线在某一点的曲率为

$$K(l) = d\theta(l)/dl \tag{8.3-47}$$

在这里有

$$K(l) = \frac{d\varphi(l)}{dl} = \frac{dt}{dl} \cdot \frac{d\varphi^*(t)}{dt} \cdot \frac{dt}{dl} \tag{8.3-48}$$

从式(8.3-34)有

$$t = \frac{2\pi l}{L} \tag{8.3-49}$$

$$dt/dl = 2\pi/L$$

代入式(8.3-48)得

$$K(t)=\frac{2\pi}{L}\left(1+\frac{d\varphi^{*}(t)}{dt}\right)=\frac{2\pi}{L}\left[1-\sum_{k=1}^{\infty}kA_k\sin(kt-\alpha_k)\right]$$

$$=\frac{2\pi}{L}\left[1+\sum_{k=1}^{\infty}kA_k\cos(kt-\beta_k)\right]$$

(8.3-50)

式中：$\beta_k=\alpha_k-\frac{\pi}{2}$。所以曲率 $K(t)$ 的傅里叶描述子为 $\left(kA_k,\ \alpha_k-\frac{\pi}{2}\right)$。由于上式中有因子 $2\pi/L$，显然 $K(t)$ 和闭合曲线的周长有关，但是若改成 $LK(t)$ 就和尺寸无关了。

2. 离散情况下的傅里叶描述子

在离散情况下，闭合曲线是由一系列连通的点组成的。设按逆时针方向曲线上的点依次为 $s_1,\ s_2,\ \cdots,\ s_N$，N 为闭合曲线上点的总数。这里的 s_i 相当于连续情况下的 l。若将这一离散的闭合曲线置于 $x-y$ 复平面上，则曲线上每一点可表示为

$$z(s_i)=x(s_i)+jy(s_i) \tag{8.3-51}$$

则曲线上点的切线向量可以用差分的形式表示

$$T(s_i)=[x(s_i)-x(s_{i-1})]+j[y(s_i)-y(s_{i-1})] \tag{8.3-52}$$

切线与 x 轴的夹角为：

$$\Phi(s_i)=\arctan\left[\frac{y(s_i)-y(s_{i-1})}{x(s_i)-x(s_{i-1})}\right] \tag{8.3-53}$$

曲线在某一点 s_i 的曲率为：

$$K(s_i)=\Phi(s_i)-\Phi(s_{i-1}) \tag{8.3-54}$$

现举例说明 $K(s_i)$ 的变化。设一等边三角形，在离散情况下，假如取其内部的点代表这个等边三角形，如图 8.16(a)所示。在沿其边界点描述时，若采用 4 连通定义，则得图 8.17(a)。一共由 14 点组成，即 $N=14$。现从 s 点开始作 $K(s_i)$，则得图 8.17(b)。如果采用 8 连通定义，则得到图 8.17(c)，这时 $N=10$，与它相对应的 $K(s_i)$ 如图 8.17(d)所示。由此得到了两种周期序列。

现在可以将以上周期序列作离散傅里叶变换，得

$$A_n+jB_n=\sum_{i=1}^{N}K(s_i)\exp\left[-j\frac{\pi ns_i}{N}\right],\ n=0,\ 1,\ 2,\ \cdots,\ N-1$$

(8.3-55)

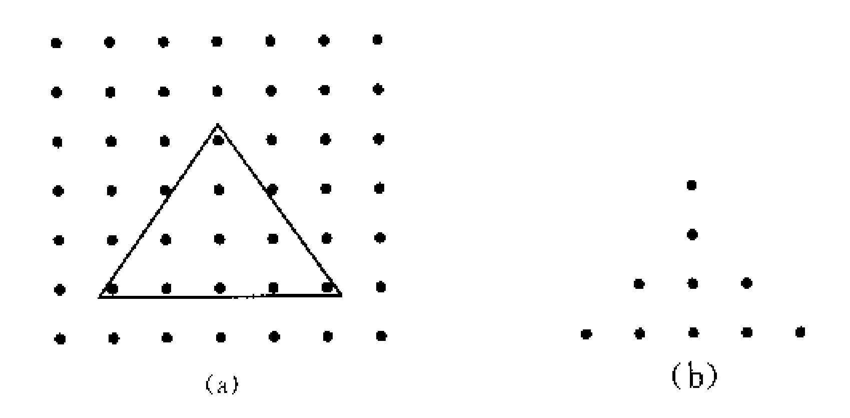
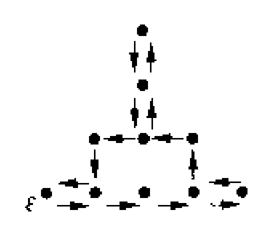
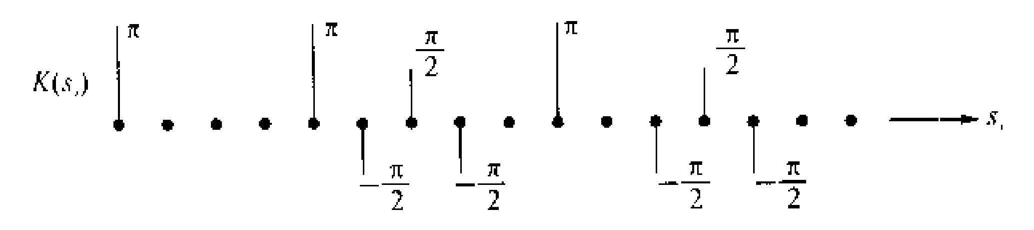


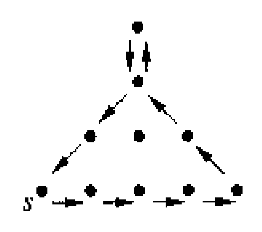
图 8.16 离散点组成的等边三角形



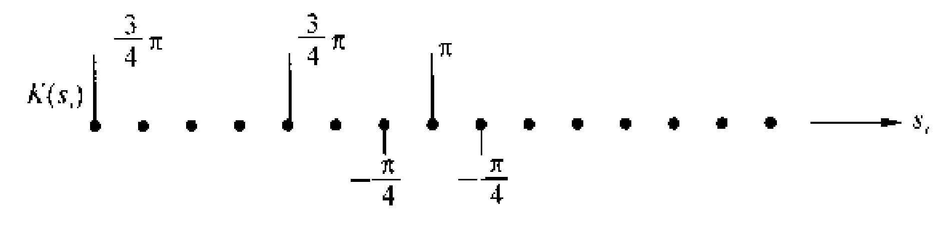
(a) 4 连通情况



(b) 4 连通情况



(c) 8 连通情况



(d) 8 连通情况

图 8.17 4 连通和 8 连通时的 $K(s_i)$

这里(A_n、B_n)就是离散情况下的傅里叶描述子。通常情况下 N 较大,故系数(A_n, B_n)较多,一般只取前面几项来描述闭合曲线的形状。

8.3.4 小波描述子

这是一种利用小波变换的多尺度平面曲线描述子。前面讲到,对于平面闭合曲线可以用傅里叶描述子来表示。由傅里叶变换的特点决定了傅里叶描述子是一种针对曲线的全局性特征的,对于曲线的局部的细节则难以正确地刻画。而小波变换由于有良好的局域性,小波描述子可以描述曲线的细节部分。另外利用小波的多分辨率分析,小波描述子可以实现多分辨率描述。

小波描述子是建立在离散周期小波变换(DPWT)的基础上的,用它的归一化DPWT系数作为平面曲线的小波描述子。下面首先介绍一维离散周期小波变换,然后介绍小波描述子和它的唯一性和其他特征。

1. 一维离散周期小波变换

参数化闭合曲线可以表示成周期序列,但是定义在 $L^2(R)$ 上的小波却不适合这种表示,因而需要引入周期小波。周期小波可以定义成连续小波作周期性平移后的总和。

设 $\phi(t)$ 是 $L^2(R)$ 上的尺度函数,$\{V_j\}$ 是 $L^2(R)$ 的多分辨率近似。对于尺度 j,$\phi(t)$ 的伸缩平移系为:

$$\{\phi_{j,n}(t) = 2^{-m/2}\phi(2^{-m}t - n);\ j,\ n \in \mathbf{Z}\} \tag{8.3-56}$$

构成了子空间 V_j 上的正交基。

对应于尺度函数 $\phi(t)$,存在一小波函数 $\psi(t)$,它的伸缩平移系为:

$$\{\psi_{j,n}(t) = 2^{-m/2}\psi(2^{-m}t - n);\ j,\ n \in \mathbf{Z}\} \tag{8.3-57}$$

构成了子空间 W_j 上的正交基。它们符合以下的双尺度方程

$$\phi(t) = \sqrt{2}\sum_n h_n\phi(2t - n) \tag{8.3-58}$$

$$\psi(t) = \sqrt{2}\sum_n g_n\phi(2t - n) \tag{8.3-59}$$

式中:

$$g_k = (-1)^k h_{1-k}$$

周期小波可以表示成连续小波周期性平移后的总和,为此,$L^2([0, 1])$空间上一维离散周期小波定义为

$$\tilde{\phi}_{j,n}(t)=\sum_{l\in Z}\phi_{j,n}(t+l),\quad j\leqslant 0;\ n=0,\cdots,2^{-j}-1 \tag{8.3-60}$$

$$\tilde{\psi}_{j,n}(t)=\sum_{l\in Z}\psi_{j,n}(t+l),\quad j\leqslant 0;\ n=0,\cdots,2^{-j}-1 \tag{8.3-61}$$

相应的周期的多分辨率近似空间为

$$\tilde{V}_j=\overline{\mathrm{span}\{\tilde{\phi}_{j,n};\ n\in Z\}}$$

$$\tilde{W}_j=\overline{\mathrm{span}\{\tilde{\psi}_{j,n};\ n\in Z\}} \tag{8.3-62}$$

而且有

$$\cdots\tilde{V}_{-2}\supset\tilde{V}_{-1}\supset\tilde{V}_0$$

以及

$$\tilde{V}_{j-1}=\tilde{V}_j\oplus\tilde{W}_j \tag{8.3-63}$$

可以证明，当 j 为负整数时，$\tilde{V}_j$ 是有限维的，它由 $2^{|j|}$ 个函数组成，$n=Z_j=\{0,\cdots,2^{-j}-1\}$，对小波函数有类似结果。

函数 $f(x)\in\tilde{V}_{J_0-1}$ 的有限尺度正交小波展开可以表示成

$$f(t)=\sum_{n=0}^{2^{-(J_0-1)}-1}C_{J_0-1}[n]\,\tilde{\phi}_{J_0,n}(t)$$

$$=\sum_{n=0}^{2^{-J_M}-1}C_{J_M}[n]\,\tilde{\phi}_{J_M,n}(t)+\sum_{j=J_0}^{J_M}\sum_{n=0}^{2^{-J_0}-1}D_j[n]\tilde{\psi}_{j,n}(t) \tag{8.3-64}$$

式中：

$$C_{J_M}[n]=\int_0^1 f(t)\phi_{J_M,n}(t)\mathrm{d}t \tag{8.3-65}$$

$$D_j[n]=\int_0^1 f(t)\psi_{j,n}(t)\mathrm{d}t \tag{8.3-66}$$

周期小波的有限尺度小波变换也有相应的快速算法。设滤波器对为

$$\tilde{g}_{j,l}=\sum_{k\in Z}g_{(l\bmod 2^{|j|})-2^{|j|}k} \tag{8.3-67}$$

$$\tilde{h}_{j,l}\equiv\sum_{k\in Z}h_{(l\bmod 2^{|j|})+2^{|j|}k} \tag{8.3-68}$$

可以证明，$\tilde{g}_{j,l}$ 和 $\tilde{h}_{j,l}$ 是周期为 2^{m} 的周期序列，则

$$C_{j+1}[n]=\sum_{l\in Z_j}\tilde{h}_j(l-2n)C_j(l) \tag{8.3-69}$$

$$D_{j+1}[n]=\sum_{l\in Z_j}\tilde{g}_j(l-2n)C_j(l) \tag{8.3-70}$$

式中：$j=J_0,\cdots,J_M-2,J_M-1$。

同样，通过合成公式可以从小波分解系数 $C_{J_M}[n]$ 和 $D_j[n]$ 求得 $D_{J_0}[n]$。

$$C_j[n]=\sum_{k\in Z_{m-1}}[\tilde{h}_j(n-2k)C_{j-1}[k]+g_j(n-2k)D_{j-1}[k]] \tag{8.3-71}$$

式中：$j=J_M-1,J_M-2,\cdots,J_0$。

式(8.3-69)和(8.3-70)称为正向离散周期小波变换，式(8.3-71)称为反向离散周期小波变换。

2. 基于坐标的小波描述子

设顺时针方向的平面闭合曲线的参数坐标为 $x(t)$ 和 $y(t)$，

$$\alpha(t)=\begin{bmatrix}x(t)\\y(t)\end{bmatrix},\ t(l)=\frac{l}{L},\ 0\leqslant l\leqslant L \tag{8.3-72}$$

式中：t 为归一化弧长；l 为从某起点 t_0 开始的弧长；L 为总的弧长。

对参数化坐标作小波变换，得

$$\begin{bmatrix}x(t)\\y(t)\end{bmatrix}=\begin{bmatrix}x_a^M(t)\\y_a^M(t)\end{bmatrix}+\sum_{j=M-M_0}^{M}\begin{bmatrix}x_d^j(t)\\y_d^j(t)\end{bmatrix} \tag{8.3-73}$$

式中：

$$x_a^M(t)=\sum_n a_n^M\tilde{\phi}_n^M(t) \tag{8.3-74}$$

$$y_a^M(t)=\sum_n c_n^M\tilde{\phi}_n^M(t) \tag{8.3-75}$$

称为在 M 尺度下的近似系数，而

$$x_d^j(t)=\sum_n r_n^j\tilde{\psi}_n^j(t) \tag{8.3-76}$$

$$y_d^j(t)=\sum_n d_n^j\tilde{\psi}_n^j(t) \tag{8.3-77}$$

称为在尺度 j 下的细节信号，$j=M-M_0$ 是最精细的尺度，$j=M$ 是粗尺度。

则式(8.3-74)～(8.3-77)中的小波系数 a_n^M，c_n^M，r_n^j 和 d_n^j 就是平面曲线的小波描述子。

3. 小波描述子的多分辨率近似

如果只使用由较粗尺度分量组成的小波系数的子集(即 $|j|$ 取较小的值)来描

述平面曲线,就可以得到曲线形状的不同分辨率表示。若用公式表示,可以将式(8.3-73)改写成

$$\begin{bmatrix}\hat{x}(k;\ t)\\ \hat{y}(k;\ t)\end{bmatrix}=\begin{bmatrix}x_a^M(t)\\ y_a^M(t)\end{bmatrix}+\sum_{j=k}^{M}\begin{bmatrix}x_d^j(t)\\ y_d^j(t)\end{bmatrix} \tag{8.3-78}$$

式中:$M-M_0\leqslant k\leqslant M+1$。具有坐标$\hat{x}(k;\ t)$和$\hat{y}(k;\ t)$的曲线就是原始曲线的一系列多分辨率近似。这一系列近似的两个极端情况是:当$k=M-M_0$时可得原始曲线;当$k=M+1$时,得到的结果是曲线的最粗尺度的近似,即$x_a^M(t)$和$y_a^M(t)$。

尺度j下的细节信号可以表示成极坐标形式。

$$\begin{bmatrix}x_d^j(t)\\ y_d^j(t)\end{bmatrix}=\sum_n\begin{bmatrix}r_n^j\\ d_n^j\end{bmatrix}\tilde{\psi}_n^j(t)=\sum_n\begin{bmatrix}\cos\theta_n^j & -\sin\theta_n^j\\ \sin\theta_n^j & \cos\theta_n^j\end{bmatrix}\begin{bmatrix}A_n^j\\ 0\end{bmatrix}\tilde{\psi}_n^j(t) \tag{8.3-79}$$

式中:

$$\theta_n^j=\arctan\left(\frac{d_n^j}{r_n^j}\right) \tag{8.3-80}$$

$$A_n^j=\sqrt{(d_n^j)^2+(r_n^j)^2} \tag{8.3-81}$$

为了平移、缩放、旋转的不变性,可以采用曲线的微分表示,如曲率,然后对其进行小波分解,得到曲线的小波描述子。但是对闭合曲线,如果用这样的截短的小波系数集重建的曲线可能是不闭合的。这里我们还是主要讨论基于坐标的小波描述子。

4. 小波描述子的特性

(1) 缩放、平移、旋转。平面曲线的缩放、平移和旋转可以通过小波系数的适当变换来表示。

对缩放,设β是曲线的缩放因子,则有

$$\begin{bmatrix}\bar{x}(t)\\ \bar{y}(t)\end{bmatrix}=\beta\begin{bmatrix}x(t)\\ y(t)\end{bmatrix}=\beta\begin{bmatrix}x_a^M(t)\\ y_a^M(t)\end{bmatrix}+\sum_{j=M-M_0}^{M}\beta\begin{bmatrix}x_d^j(t)\\ y_d^j(t)\end{bmatrix} \tag{8.3-82}$$

由于小波变换是线性变换,由此可以得到缩放后的小波描述子

$$\begin{bmatrix}\bar{a}_n^M\\ \bar{c}_n^M\end{bmatrix}=\beta\begin{bmatrix}a_n^M\\ c_n^M\end{bmatrix} \tag{8.3-83}$$

$$\begin{bmatrix}\bar{r}_n^j\\ \bar{d}_n^j\end{bmatrix}=\beta\begin{bmatrix}r_n^j\\ d_n^j\end{bmatrix} \tag{8.3-84}$$

对于极坐标形式，有

$$\ddot{\theta}_n^j = \arctan\left(\frac{\beta d_n^j}{\beta r_n^j}\right) = \arctan\left(\frac{d_n^j}{r_n^j}\right) = \theta_n^j \tag{8.3-85}$$

$$\ddot{A}_n^j = \sqrt{(\beta d_n^j)^2 + (\beta r_n^j)^2} = \beta A_n^j \tag{8.3-86}$$

对平移，设曲线平移了 $b = (b_x, b_y)$，根据周期小波的性质，由公式(2.1)，可得

$$\int_0^1 \tilde{\psi}_n^j(t)\mathrm{d}t = \sum_{l\in Z}\int_l^{l+1}\psi(t')\mathrm{d}t' = 0 \tag{8.3-87}$$

由此，容易得到

$$\int_0^1 [f(t)+b]\tilde{\psi}_n^j(t)\mathrm{d}t = \int_0^1 f(t)\psi_n^j(t)\mathrm{d}t \tag{8.3-88}$$

可见，细节信号 $x_d^j(t)$ 和 $y_d^j(t)$ 的小波系数 r_n^j 和 d_n^j 是移不变的。平移仅仅对近似信号产生影响。因为对周期小波的尺度函数，有

$$\int_0^1 \tilde{\phi}_n^M(t)\mathrm{d}t = \sum_{l\in Z}\int_0^1 \phi(t+l)\mathrm{d}t = \int_{-\infty}^{\infty}\phi_n^M(t)\mathrm{d}t \tag{8.3-89}$$

另外，由 $\phi_n^M = 2^{-M/2}\phi(2^{-M}t - n)$，可得

$$\int_0^1 \tilde{\phi}_n^M(t)\mathrm{d}t = 2^{-M/2}\int_{-\infty}^{\infty}\phi(t)2^M\mathrm{d}t = 2^{M/2} \tag{8.3-90}$$

所以

$$\int_0^1 [f(t)+b]\,\tilde{\phi}_n^M(t)\mathrm{d}t = \int_0^1 f(t)\,\tilde{\phi}_n^M(t)\mathrm{d}t + 2^{M/2}b \tag{8.3-91}$$

由此可得平移后的小波描述子，

$$\begin{bmatrix}\ddot{a}_n^M\\ \ddot{c}_n^M\end{bmatrix} = \begin{bmatrix}a_n^M + 2^{M/2}b_x\\ c_n^M + 2^{M/2}b_y\end{bmatrix} \tag{8.3-92}$$

$$\begin{bmatrix}\ddot{r}_n^j\\ \ddot{d}_n^j\end{bmatrix} = \begin{bmatrix}r_n^j\\ d_n^j\end{bmatrix} \tag{8.3-93}$$

对旋转，设绕曲线的质心沿反时针方向旋转 φ 角，则小波描述子为

$$\begin{bmatrix}\ddot{a}_n^M\\ \ddot{c}_n^M\end{bmatrix} = \begin{bmatrix}\cos\varphi & -\sin\varphi\\ \sin\varphi & \cos\varphi\end{bmatrix}\begin{bmatrix}a_n^M\\ c_n^M\end{bmatrix} \tag{8.3-94}$$

$$\begin{bmatrix} \tilde{r}_n^m \\ \tilde{d}_n^m \end{bmatrix} = \begin{bmatrix} \cos\varphi & -\sin\varphi \\ \sin\varphi & \cos\varphi \end{bmatrix} \begin{bmatrix} r_n^m \\ d_n^m \end{bmatrix} \tag{8.3-95}$$

若用极坐标表示，则有更简单的形式

$$\tilde{\theta}_n^j = \theta_n^j + \varphi \tag{8.3-96}$$

$$\tilde{A}_n^j = A_n^j \tag{8.3-97}$$

（2）其他特性：

① 非收缩性：一般情况下，多分辨率表示的曲线与一个平滑函数多次卷积后，曲线的周长会缩短。这是因为曲线的低频和高频分量减少的缘故。而对于小波描述子则没有收缩问题。

② 不变性：指两个同样形状的曲线应该有相同的表示。

③ 唯一性：指具有不同形状的两个曲线有不同的表示。

④ 稳定性：指曲线之间小的差异应该与表示之间的差异相对应。

特性②、③、④对一般的形状描述子都应该满足。经证明，小波描述子同样满足这些特性。

8.4 区域的描述

区域是图像中宽度大于1的连通的点组成的图形。上节讨论的线条可以看作是区域的特殊情况，即宽度为1的区域。图像中要分析的对象往往是具有一定形状的区域。因此，用一定的方式来表示或描述区域就显得十分重要。由于区域的边界包含了大量有关区域形状的信息，所以用上一节讨论的傅里叶描述子和链码等也可以描述区域，但这些方法的着眼点是边界线。本节将介绍另外一些区域的描述方法，它们更多地着眼于区域本身。

本节的讨论以二值图为对象。

8.4.1 中轴变换

中轴变换就是将区域骨架化，在一些场合也作为细化的一种方法。但是由于中轴变换除了可以将区域细化成骨架外，还附带区域边界，也就是区域形状和大小的信息。它除了可以用中轴（骨架）来表示区域外，还可以由中轴变换的表示重建原始区域。因此，中轴变换也常用作区域的一种描述方法。

我们先用一种形象化的说法来说明中轴变换是怎么回事。将区域S设想成下

草原，而将 S^c 设想成无法燃烧的沼泽地。如果在 S 的边界上同时点火，则大火将烧向 S 的中心。假设，大火推进的速度相同，则四周的大火在向中心推进时会在某些地方会合，然后熄灭，直至 S 中的干草原全部烧光。以图 8.18(a)为例，当大火从四周向中央推进时，相邻两边的火头首先相遇并随之熄灭，上下两股火头最后在中央的水平线上相遇并熄灭。这样形成的熄火线就是原矩形的骨架或中轴。而对图 8.18(b)那样的圆形区域，它的骨架就是圆心一个点。从以上的说明可以看出，中轴是这样的一些点的集合，即它们和两个最近边的距离相等。因此，中轴上每个点还附带了一个信息，即它到两个边界点的距离相等。据此可以方便地重建原来的区域。中轴变换这样形象的说明[10]很好地反映了中轴变换的实质。

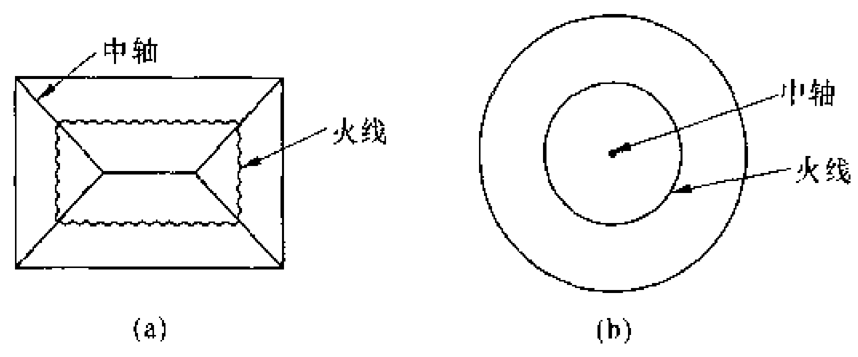


图 8.18 中轴变换

8.1.2 小节中定义了点 P 到点集 A 的距离为 P 到 A 中最近点的距离。设 S 的边界点的集合为 B，显然在 S 中一定存在一些特殊点 x，它至少与 B 中两个点 y 和 z 的距离相等，即有

$$d(x, B) = d(x, y) = d(x, z) \tag{8.4-1}$$

这些特殊点组成了 S 的中轴。这与本节一开始的说法是一致的。因为火从边界同时向中央推进，且速度相等，因此火头相遇点到边界的距离一定相等。可以证明，上述定义还可以表示为：S 的中轴上的点是与 S^c 的距离有局部极大值的点。在具体算法中，就是通过求距离的局部极值点来实现中轴变换的。因此中轴变换的结果显然与所用的距离的度量方法有关。

为方便起见，对于中轴上的点，定义一个函数

$$q(x) = d(x, B) \tag{8.4-2}$$

称为熄火函数。任一区域 R，经中轴变换后一定可以得中轴 $m(x)$ 和熄火函数 $q(x)$。它们完整地描述了这一区域，而且可以从(m, q)重建原来的区域。重建的方法很简单，以骨架上每一点为圆心，以 $q(x)$ 为半径作圆盘，所有圆盘的并集就是原来的区域 R[12]。

对中轴变换作适当处理，还可以用比较简单的形式近似地描述这一区域。为

此，定义一个形象化的量，即在熄火线上火头推进的速度。设火从边界向中央推进时的速度为1，则在熄火线上任一点 x，火头推进的速度为

$$V(x) = \mathrm{d}S/\mathrm{d}q \tag{8.4-3}$$

式中：S 是沿骨架到 x 点的距离。以图 8.18(a) 为例，根据以上定义，很显然，对四根斜线骨架来说，$V(x)$ 都为 $\sqrt{2}$，而 e 上每点的速度都为无穷大。这是因为上下两股火头在同一瞬时相遇于 e。如果取速度门限为 1.5，即 $V(x) < 1.5$ 的骨架消去，则只剩下 e，如图 8.19 所示。这与细化的结果类似。若按这种近似的表示重建，则重建后的区域如图 8.19 中实线所示。

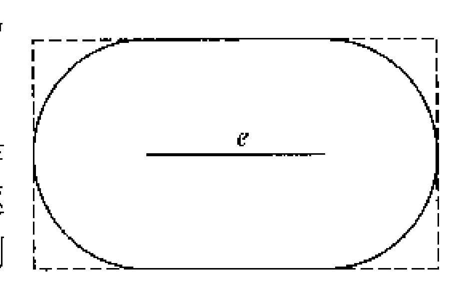
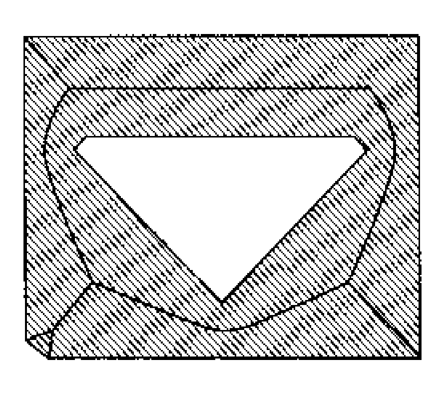


图 8.19　近似的中轴表示

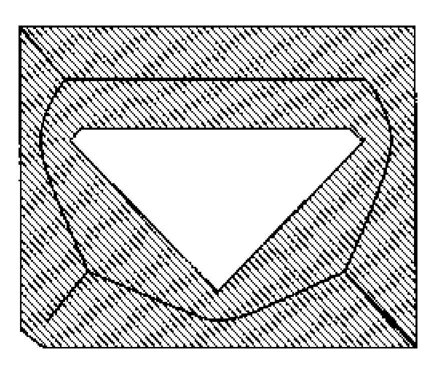
如果区域 R 中某相邻两边的夹角为 2θ，则很容易看出

$$V(x) = \mathrm{d}S/\mathrm{d}q = 1/\sin\theta \tag{8.4-4}$$

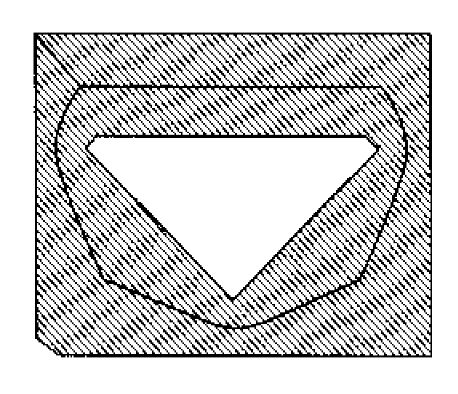
显然，夹角愈大，对应骨架点上点的速度 $V(x)$ 愈小。因此，丢弃小速度的骨架意味着忽略那些角度很大的近似直线的边界变化。图 8.20 是一种比较复杂的例子。



(a) 骨架(中间为孔)



(b) 门限 1.25



(c) 门限 1.5

图 8.20　复杂区域的中轴及近似表示

中轴变换对边界噪声十分敏感，如图 8.21 所示，克服的方法是预先对区域的边界进行平滑。

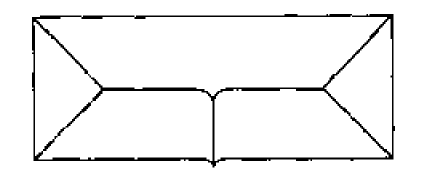
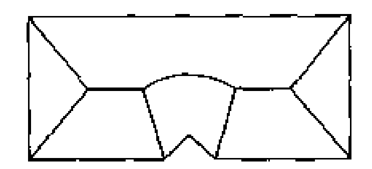
 

图 8.21 噪声对中轴变换的影响

8.4.2 矩描述

1. 矩

矩是一种线性特征，可以用来对区域进行描述。f 的线性特征可以写成 $\iint hf$，对离散函数可写成 $\sum\sum hf$ [11]。对矩来说 h 取作 $x^i y^j$。因此对二维连续函数，$(i+j)$ 阶矩为

$$m_{ij}=\int_{-\infty}^{+\infty}\int_{-\infty}^{+\infty} x^i y^j f(x, y)\mathrm{d}x\mathrm{d}y,\ i,\ j=0,\ 1,\ 2\cdots \tag{8.4-5}$$

对离散图像函数 $(i+j)$ 阶矩可表示为

$$m_{ij}=\sum_x\sum_y x^i y^j f(x, y),\ i,\ j=0,\ 1,\ 2\cdots \tag{8.4-6}$$

若以左上角为原点，行方向为 x 方向，则对 3×3 的图像可求得 $i, j\leqslant 2$ 的各阶矩如下：

3	3	2	i	0	1	0	1	2	0
2	1	1	j	0	0	1	1	0	2
1	0	1	m_{ij}	14	8	12	7	12	20

如果将图像灰度看做质量，则可以给各阶矩以明确的物理解释。例如 m_{00} 是 f 的总质量，m_{02} 和 m_{20} 是 f 绕 x 轴和 y 轴的惯量，等等。

根据唯一性定理[13]，若 $f(x, y)$ 是分段连续的，即只在 xy 平面的有限区域有非零值，则所有各阶矩都存在，而且矩系列 $\{m_{ij}\}$ 唯一地被 $f(x, y)$ 所确定。反之 $\{m_{ij}\}$ 也唯一地确定了 $f(x, y)$。因此矩特征可以用来描述图像中的区域。

与式(8.4-6)相应的 $i+j$ 阶中心矩为

$$\mu_{ij}=\sum_x\sum_y (x-\bar{x})^i(y-\bar{y})^j f(x, y) \tag{8.4-7}$$

式中：

$$\bar{x}=\frac{m_{10}}{m_{00}},\ \bar{y}=\frac{m_{01}}{m_{00}} \tag{8.4-8}$$

为 f 的矩心。

根据式(8.4－7)可以导得三阶以内的中心矩[11]如下。

$$
\begin{aligned}
&\mu_{00}=m_{00} && \mu_{11}=m_{11}-\bar{y}m_{10}\\
&\mu_{10}=0 && \mu_{30}=m_{30}-3\bar{x}m_{20}+2m_{10}\bar{x}^2\\
&\mu_{01}=0 && \mu_{12}=m_{12}-2\bar{y}m_{11}-\bar{x}m_{02}+2\bar{y}^2m_{10}\\
&\mu_{20}=m_{20}-\bar{x}m_{10} && \mu_{21}=m_{21}-2\bar{x}m_{11}-\bar{y}m_{20}+2\bar{x}^2m_{01}\\
&\mu_{02}=m_{02}-\bar{y}m_{01} && \mu_{03}=m_{03}-3\bar{y}m_{02}+2\bar{y}^2m_{01}
\end{aligned}
\tag{8.4-9}
$$

用零阶中心矩对其余各阶中心矩规格化，可得规格化中心矩

$$\eta_{ij}=\mu_{ij}/\mu_{00}^{r} \tag{8.4-10}$$

式中：

$$r=\frac{i+j}{2},\quad i+j=2,3$$

为了使矩描述子与平移、大小、旋转等无关，可以利用二阶矩和三阶矩导出下面七个不变矩组[12]。

$$
\begin{aligned}
\phi_1&=\eta_{20}+\eta_{02}\\
\phi_2&=(\eta_{20}-\eta_{02})^2+4\eta_{11}^2\\
\phi_3&=(\eta_{30}-3\eta_{12})^2+(3\eta_{21}-\eta_{03})^2\\
\phi_4&=(\eta_{30}+\eta_{12})^2+(\eta_{21}+\eta_{03})^2\\
\phi_5&=(\eta_{30}-3\eta_{12})(\eta_{30}+\eta_{12})[(\eta_{30}+\eta_{12})^2-3(\eta_{21}+\eta_{03})^2]\\
&\quad+(3\eta_{21}-\eta_{03})(\eta_{21}+\eta_{03})[3(\eta_{30}+\eta_{12})^2-(\eta_{21}+\eta_{03})^2]\\
\phi_6&=(\eta_{20}-\eta_{02})[(\eta_{30}+\eta_{12})^2-(\eta_{21}+\eta_{03})^2]+4\eta_{11}(\eta_{30}+\eta_{12})(\eta_{21}+\eta_{03})\\
\phi_7&=(3\eta_{21}-\eta_{03})(\eta_{30}+\eta_{12})[(\eta_{30}+\eta_{12})^2-3(\eta_{21}+\eta_{03})^2]\\
&\quad+(3\eta_{12}-\eta_{30})(\eta_{21}+\eta_{03})[3(\eta_{30}+\eta_{12})^2-(\eta_{21}+\eta_{03})^2]
\end{aligned}
\tag{8.4-11}
$$

2. 主轴

前面说过，m_{02}和m_{20}分别是绕x轴和y轴的惯量。设一直线过点(α, β)，且与x轴夹角为θ，即直线上的点满足

$$(x-\alpha)\sin\theta-(y-\beta)\cos\theta=0 \tag{8.4-12}$$

则绕这根线的惯量为：

$$\sum\sum[(x-\alpha)\sin\theta-(y-\beta)\cos\theta]^2 f(x, y) \tag{8.4-13}$$

显然可以找到一条直线，使绕它的惯量最小。现求使上式最小时的α、β和θ。将式(8.4-13)对α或β求导，并令其等于0，得

$$\sum\sum[(x-\alpha)\sin\theta-(y-\beta)\cos\theta]f(x, y)=0$$

或写成

$$m_{10}\sin\theta-\alpha m_{00}\sin\theta-m_{01}\cos\theta+\beta m_{00}\cos\theta=0$$

用m_{00}除上式得

$$(\bar{x}-\alpha)\sin\theta-(\bar{y}-\beta)\cos\theta=0 \tag{8.4-14}$$

从上式可以得出，使式(8.4-13)最小的α、β为：

$$\alpha=\bar{x}, \beta=\bar{y} \tag{8.4-15}$$

因此，关于最小惯量线的惯量为：

$$\sum\sum[(x-\bar{x})\sin\theta-(y-\bar{y})\cos\theta]^2 f(x, y) \tag{8.4-16}$$

显然，最小惯量线是过矩心$(\bar{x}, \bar{y})$且与x轴夹角为θ的直线，又称为f的主轴。为了找出主轴与x轴的夹角θ，将坐标原点移至矩心$(\bar{x}, \bar{y})$。这时主轴方程为$y=x\tan\theta$，而关于主轴的惯量为：

$$\sum\sum(x\sin\theta-y\cos\theta)^2 f(x, y)=\bar{m}_{20}\sin^2\theta-2\bar{m}_{11}\sin\theta\cos\theta+\bar{m}_{02}\cos^2\theta \tag{8.4-17}$$

式中$\bar{m}_{20}$，$\bar{m}_{11}$，$\bar{m}_{02}$是坐标原点位于矩心$(\bar{x}, \bar{y})$时的二阶矩。将上式对θ求导并令其等于0，得

$$2\bar{m}_{20}\sin\theta\cos\theta-2\bar{m}_{11}(\cos^2\theta-\sin^2\theta)-2\bar{m}_{02}\sin\theta\cos\theta=0$$

或写成

$$\bar{m}_{20}\sin2\theta-2\bar{m}_{11}\cos2\theta-\bar{m}_{02}\sin2\theta=0$$

由此得

$$\tan 2\theta = \frac{2m_{11}}{\bar{m}_{20} - \bar{m}_{02}} \tag{8.4-18}$$

$$\theta = \frac{1}{2}\arctan\left(\frac{2\bar{m}_{11}}{\bar{m}_{20} - \bar{m}_{02}}\right) \tag{8.4-19}$$

即主轴与 x 轴夹角由上式确定。而且 f 的绕主轴的惯量最小。

利用主轴的概念可以对图像 f 进行旋转规格化，即将 f 旋转一定的角度，使主轴位于某个标准的方向，比如说位于垂直方向。

主轴可以看做是与 f 最佳配合的线。在一般情况下，可以找到一个次数较高的曲线，使它在不同的意义上与 f 配合最佳。例如，一个二次曲线

$$q(x, y) = ax^2 + bxy + cy^2 + ux + vy + w = 0 \tag{8.4-20}$$

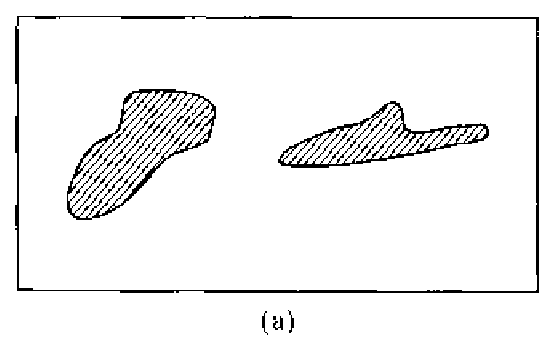
然后可以设法求出系数 a, b, c, u, v, w，使下式达到最小。

$$\sum\sum(q(x, y))^2 f(x, y) \tag{8.4-21}$$

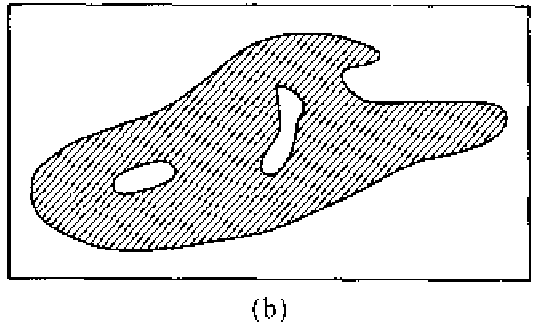
这样求出的曲线可以看做 f 的一种二次型主轴。同样，也可以通过坐标变换实现相应于这样曲线的形状的规格化。

8.4.3 拓扑描述子

拓扑特征是对图像中区域结构形状的总体描述。它的特点是不受所谓"橡皮被单"(Rubber Sheet)式畸变的影响。就是说当图形由于拉伸、压缩、扭曲等而变形时，拓扑特性不变，但不能有撕裂和折叠。因此，诸如距离、面积、周长、平行、垂直等与形状有关的特性和概念都不属于拓扑特性。



(a)



(b)

图 8.22 连通分量数和孔数

最普通的一个拓扑特征是连通分量的个数。S 的连通分量是 S 的子集，它们每一个都是符合连通性定义的点的最大集合。图 8.22(a)有 2 个连通分量。

另一个有用的拓扑特征是孔的个数。图 8.22(b)是一个有 2 个孔的图像。显

然，连通数和孔数是不受“橡皮被单”式畸变的影响的。

设连通分量数为 C，孔数为 H，则可以定义另一个拓扑特性——欧拉数 E。

$$E = C - H \tag{8.4-22}$$

图 8.22(a)(b)的欧拉数分别为 2 和 −1。

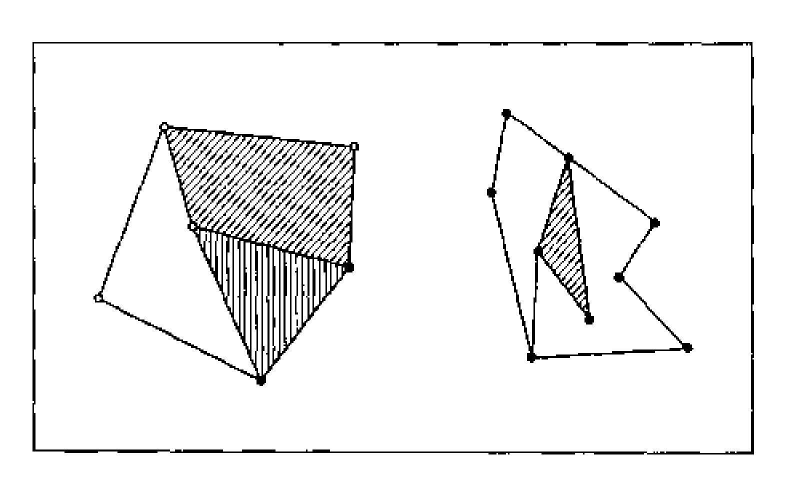


图 8.23 网络图形的拓扑特性

下面看一种由直线段组成的图形，如图 8.23 所示。这种图形的连通分量仅由细线组成。这些线段组成了各种多边形网络。网络的内部区域常分为两类，即面和孔。设多边形网络的面数为 F，孔数为 H，连通分量数 C，顶点数为 V，边数为 S，则欧拉公式有如下形式

$$V - S + F = C - H = E \tag{8.4-23}$$

图 8.23 所示图形有 2 个连通分量，3 个孔，3 个面、15 个顶点，19 条边，由式(8.4-23)可求得这个图像的欧拉数为

$$E = 15 - 19 + 3 = 2 - 3 = -1$$

拓扑描述子反映了图像中区域的总体的综合特征。在图像分析中常用作一种初步分类的参数，或作为其他描述方法的辅助描述。例如，在文字识别中，孔数常作为文字形状描述的一个特征。

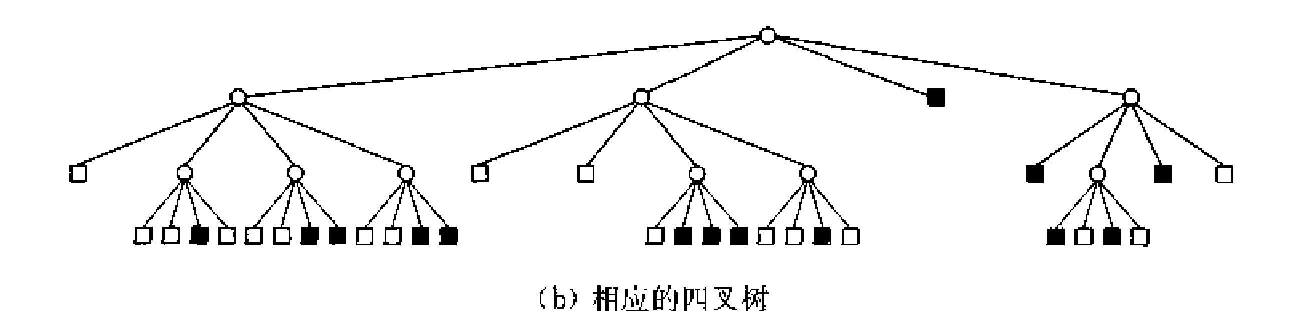
8.4.4 四叉树

用四叉树可以很好地描述图像[17]。它在图像处理、景象分析和计算机图形学等方面有很大的用途[18]。

四叉树的形成是一个金字塔式的观察和处理的过程。它是通过反复地四分图像得到的。为了能将图像反复四分，要求图像大小为 $2^k \times 2^k$，k 为正整数。首先将整幅图像看作是根节点。如果全图仅由单个值组成(0 或 1)，则四叉树到此结束，

0	0	0	0	0	0	0	0
0	0	1	0	0	0	0	0
0	0	0	0	0	1	0	0
1	1	1	1	1	1	1	0
1	1	1	1	1	1	1	0
1	1	1	1	1	1	1	0
1	1	1	1	1	1	0	0
1	1	1	1	1	1	0	0

(a) 8×8 大小的二值图



(b) 相应的四叉树

图 8.21 四叉树表示法

否则将全图分成四块，每块分别用一个节点表示。显然，根据每块中灰度值的分布可能有三种情况，即全都为 0、全都为 1、既有 0 又有 1。相应地，用三类节点来表示。对第一和第二种情况，分别用白节点(0 节点)和黑节点(1 节点))表示，它们又称叶节点。对第三种情况用灰节点表示，它们属于中间节点。将属于中间节点的子图块再四分，又可得到它的 4 个子节点。如此反复进行直至所有节点都为黑或白，即不存在没有后代节点的中间节点。为了表示金字塔的等级，以根节点为 0 层，然后对每个等级的四分增加一层，即从根节点开始依次为 0，1，2，…层。对于 $2^k \times 2^k$ 的图像，最底层最多为第 k 层。而第 n 层(如果存在的话)的子图块的大小为 2^{k-n}。如果第 k 层存在，则它的叶节点对应于单个像素。

图 8.24 是四叉树的例子。四叉树作为区域描述的主要缺点是描述的结果随图形的移动而变。即使两幅图像仅仅平移了一个像素，所得结果也可能很不相同。这是与前述一些方法不同的地方。

在实际应用中，为了节省存储量，便于计算机作其他的处理操作等原因，需要用适当的数据结构和编码方法来表示四叉树。这里举两种为例。一种是节点描述法，将节点分为三类，分别用符号{(，0，1}表示。其中“(”表示中间节点，“0”为黑节点，“1”是白节点。然后用深度优先搜索方法(Depth First Search)依次记录所有节点，即从根节点开始沿最左的通路向下搜索，直至叶节点。然后回溯到上一代节

点,再向下搜索。如此循环直至全树搜索完毕。现以图 8.25 简单的四叉树为例,得这棵树的数据(符号)表示为

$$((11100(0011((1000111$$

另外一种表示为叶节点表示法,又称"深度-颜色"表示法。即将每一个叶节点用它所处的层次和它代表的颜色来表示。表示的顺序为从左到右依次记录。对图 8.25 所示四叉树记录结果为

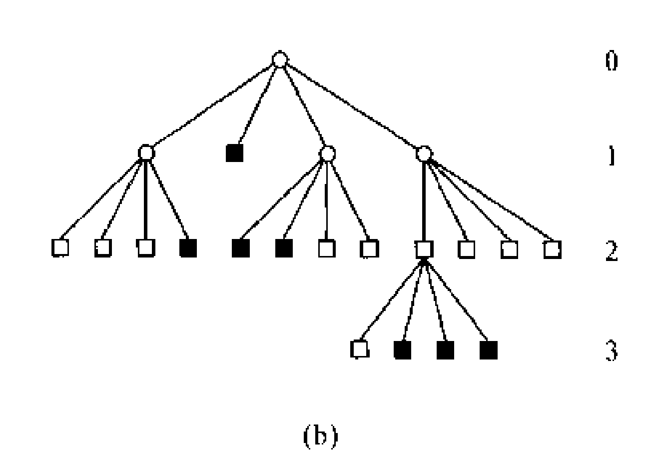
$$(2, 1)\ (2, 1)\ (2, 1)\ (2, 0)\ (1, 0)\ (2, 0)\ (2, 0)\ (2, 1)\ (2, 1)$$

$$(3, 1)\ (3, 0)\ (3, 0)\ (3, 0)\ (2, 1)\ (2, 1)\ (2, 1)$$

据分析[19],这两种方式所需存储量等数量级,但后一种方法有利于对四叉树表示的图像作进一步的处理。

0	0	0	0	1	1	1	1
0	0	0	0	1	1	1	1
0	0	1	1	1	1	1	1
0	0	1	1	1	1	1	1
1	1	1	1	1	0	0	0
1	1	1	1	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0

(a)



(b)

图 8.25 一种简单的四叉树

四叉树描述方法与其他方法相比各有自己的特点,例如,四叉树描述简单,容易进行集合运算等。而且在四叉树和其他描述方法之间还可以直接进行转换,例如四叉树和边界描述之间的相互转换,参考文献[20, 21]对此作了详细的论述。

8.4.5 行程编码

行程编码是图像编码的一种,也可以用来作区域描述。图像中每一行都由一系列具有一定灰度值的点组成,即为$\{x_i\}$, $i=0, 1, \cdots, N-1$。如果把这些点中相邻点灰度值保持不变的那些点称作为行程,则一行图像就完全由这些行程的长度和灰度值所决定。也就是可以把点的离散序列$\{x_i\}$映射为一个整数对(g_k, l_k)序列。g_k是行程的灰度值。l_k是它的长度,$k=1, 2, \cdots, m$,为行程的个数。这就是所谓行程编码。在行程数很少的情况下,行程编码可以作为区域描述的一个经济

而合理的方法;对一幅实际图像来说,每一行的行程数最少为1,最多为N。前者一行中所有像素的值都相同,后者在一行中则没有任何两个相邻点的值相同。相应的行程长度分别为N和1。若用普通的二进制码存放(g_k, l_k)序列,设灰度级为M,一行的总长为N,一行中的行程数为m,则对g_k需$\log_2 M$比特,l_k需$\log_2 N$比特。描述整个行则需要

$$m(\log_2 M + \log_2 N) \tag{8.4-24}$$

比特。而存储原图的每一行需$N\log_2 M$比特。显然只有当$m \ll N$时,这种描述方法才是可取的。

以上是对每行进行描述。对整幅图像最简单的方法是逐行作类似的表示,如图8.26所示。表示结果呈以下形式:

$$(g_{11}, l_{11})(g_{12}, l_{12})\cdots(g_{1m_1}, l_{1m_1})$$

$$(g_{21}, l_{21})(g_{22}, l_{22})\cdots(g_{2m_2}, l_{2m_2})$$

$$\vdots$$

$$(g_{k1}, l_{k1})(g_{k2}, l_{k2})\cdots(g_{km_k}, l_{km_k})$$

其中,k为总行数。

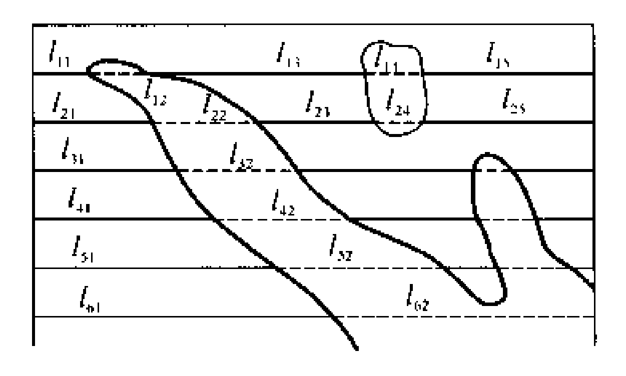


图8.26 行程编码

对于二值图,只有两个可能的值。在一行中,相邻两个行程的值一定是交替出现的,因此只需在第一个行程中标出它的值。这样,一行的行程编码变为:

$$g_1 l_1 l_2 \cdots l_m$$

g_1只有两个值($M=2$),只需1比特。这时描述一行的行程编码需要

$$1 + m\log_2 N \tag{8.4-25}$$

比特,用普通表示方法则为N比特。

当行程数很少时,尤其是对于只有较少区域的二值图,行程编码的描述方法简

单方便,描述和重建的算法都很简单。

在图像数据压缩中也可以采用行程编码,这时可以根据各种行程长度在图像中出现的概率用不同长度的码对其进行编码,以提高数据的压缩率。

8.5 几何特征的测量

这一节介绍一些几何特征的测量方法。例如计数、面积、周长、投影等等。这些参数的测量在某些领域的实际应用中十分重要。一幅经过分割处理以后的二值图像由集合 S 和 $\bar{S}$ 组成。S 的各个连通分量可以看做一个个的物体,这些通常是需要进行分析和测量的对象。为方便起见,称这些连通分量为物体,$\bar{S}$ 称之为背景。

8.5.1 拓扑特征及物体计数

1. 连通分量的标注

为了把 S 的各个连通分量表示成不同的物体,常给不同分量以不同的符号(或数值),这就是连通分量的标注。

标注算法一般通过对图像作两次扫描来实现。下面分别讨论 4 连通分量和 8 连通分量的标注方法。

对 4 连通分量,在第一次扫描中,遇到灰度值为 1 的点 P,则检查 P 点的左面及上面两个邻点。如果存在这样的邻点(P 为第一行第一列者除外),则一定已经被标注。因此可能有三种情况:

(1) 两个邻点都为 0,则给 P 以新的标记。

(2) 两个邻点中有一个为 0,则 P 的标记与另一个不为 0 的邻点的标记相同。

(3) 两个邻点都不为 0,则 P 的标记与它左面邻点的标记相同,并记下两个邻点的标记等价。

第一次扫描后所有等于 1 的点都被标注,但是同一个连通分量可能有多个符号。为此,必须对已记录的符号的等价情况进行分类,所有等价的符号归为一类,然后在第二次扫描中根据分类情况,将同一类改用同一符号标注。图 8.27 是标注算法的简单例子。

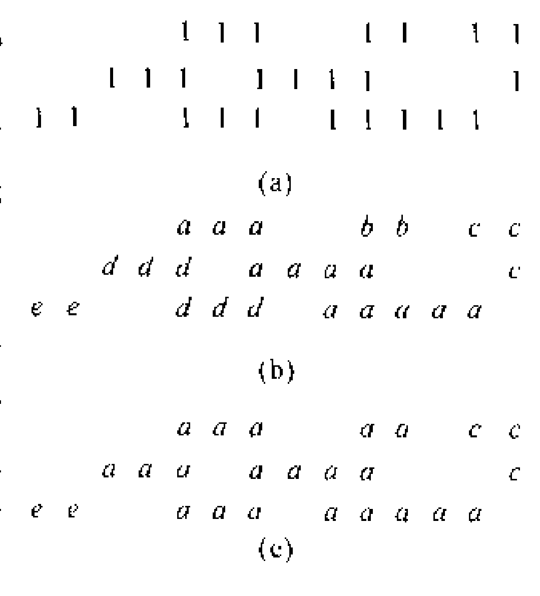


图 8.27 连通分量的标注(4 连通)

8 连通情况的算法类似,只是第一次标注

时略有不同。这时，对值为 1 的 P 点要检查左、上、左上、右上 4 个邻点。也有三种情况：

(1) 全部为 0，给 P 以新的符号。

(2) 其中有一个邻点为 1，则给 P 以同样的符号。

(3) 如果 2 个或 2 个以上的邻点为 1，则取其中一个的符号给 P，并记录所有邻点符号的等价性。

完成以上的初始标注及符号的等价记录以后，其他步骤同 4 连通的情况。

以上算法是直接从图像的阵列表示对图像中连通分量进行标注的。类似的标注方法可以推广到图像的其他表示，例如由图像的行程编码、四叉树和中轴变换等对图像进行标注[11]。

2. 计数

物体的数目就是连通分量的个数，如果图像已经标注，则所用符号数就是物体的个数。这在标注过程中就可完成。本小节介绍两种未经标注的图像的计数方法。

首先介绍一种适用于单连通物体(无孔存在)的计数法。令 v 为图像中 $\begin{matrix}0 & 0\\1 & 1\end{matrix}$ 的个数，e 为 $\begin{matrix}0 & 1\\1 & 1\end{matrix}$ 的个数，d 为 $\begin{matrix}1 & 0\\0 & 1\end{matrix}$ 的个数。则对于 S 为 4 连通，S' 为 8 连通的情况，物体的个数为

$$n = v - e + d \tag{8.5-1}$$

这种算法可以理解为向上凸的端点数减去向下凹的谷点数，即凸的端点为 1，凹的谷点为 −1，如图 8.28 所示。

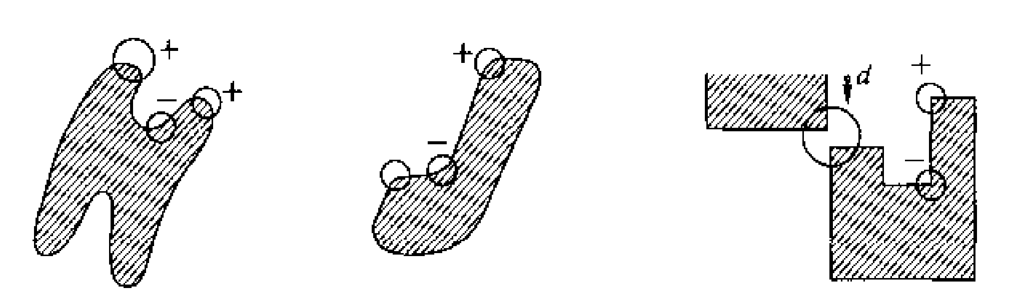


图 8.28 单连通物体计数，○⁺ 为凸，○⁻ 为凹

下面介绍一种适用于一般情况的计数方法。

这是一种特殊的收缩算法。一般的收缩算法会破坏区域的连通性，不能直接用来计数。若用类似细化的收缩方法，只收缩那些简单的边界点，那么经过反复迭代后，一个单连通区域最后将只剩下一个点而后消失。当然只要改进一下算法，可

使最后剩下的一点不被消去，以此对单连通物体计数。但是对复连通情况，因为有孔的存在，最后不能得到单个点，无法计数。为此必须采用一种特殊的收缩算法，以适应一般情况的物体计数。

设 S 为 4 连通，S' 为 8 连通，且将 P 点的三个邻点记作 X、Y、Z。

$$\begin{matrix} P & X \\ Y & Z \end{matrix}$$

定义一个运算 Ψ，这个运算当 P 为 1，X、Y 都为 0 时置 P 为 0；当 P 为 0，X、Y、Z 都为 1 时，置 P 为 1，即

$$\Psi(P)=\begin{cases}0, & P=1\text{ 且 }X=Y=0\\ 1, & P=0\text{ 且 }X=Y=Z=1\end{cases} \tag{8.5-2}$$

可以证明 Ψ 运算不会破坏区域的连通性。对 S 的任意连通分量 C，反复运用 Ψ 运算后，C 将缩成一点 (x_c, y_c)，x_c 为 C 中最左点的 x 坐标，y_c 为 C 中最上面点的 y 坐标。就是说，这一点趋向于 C 的西北方向，而且即使 C 中有孔存在也是如此。这时若不对孤立点作特殊标记，则下一步收缩将使其消失。为此在作 Ψ 运算时应对孤立点作出特殊处理。

式(8.5-2)所示 Ψ 运算的例子示于图 8.29。

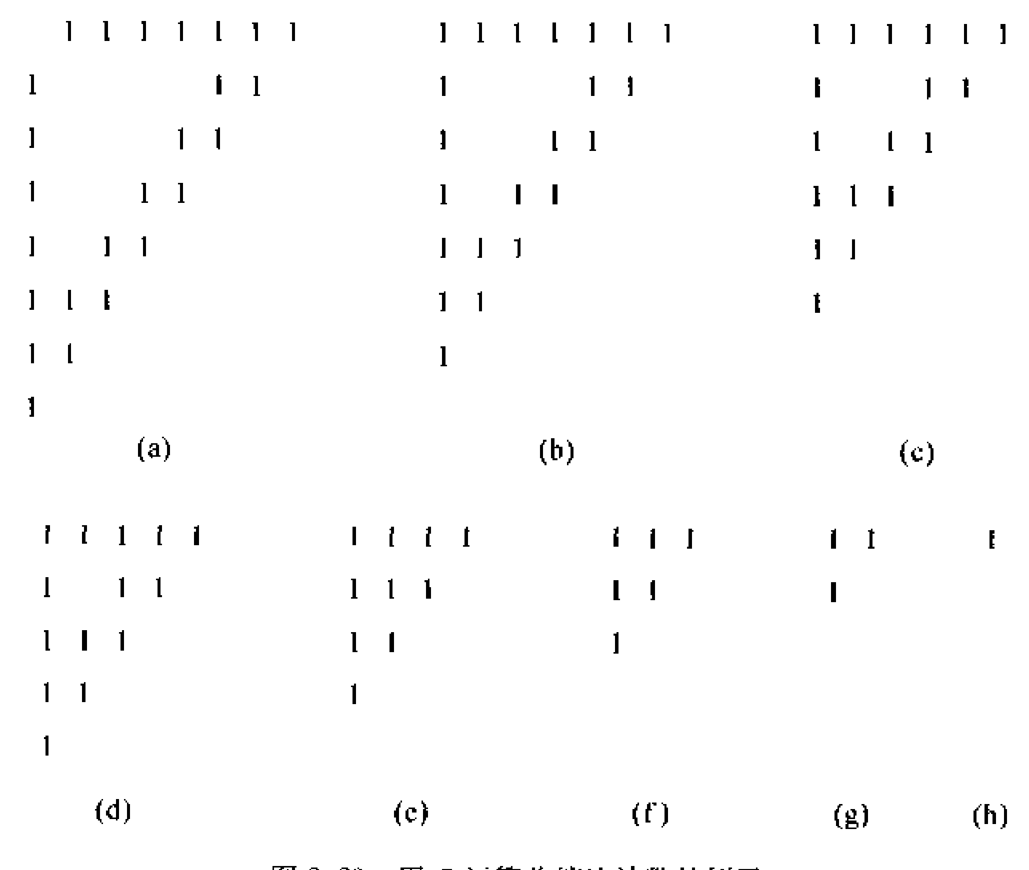


图 8.29 用 Ψ 运算收缩法计数的例子

若 S 为8连通，S^c 为4连通，则 Ψ 运算可定义为：

$$\Psi(P)=\begin{cases}0, & P=1\text{ 且 }X=Y=Z=0\\ 1, & P=0\text{ 且 }X=Y=1\end{cases} \tag{8.5-3}$$

上面定义的 Ψ 运算中，P 若采用其他的 2×2 邻域，则 Ψ 运算将使收缩向其他方向发展，例如东北、东南、西南等。

前面说过，欧拉数 E 是重要的拓扑特性，它等于连通分量数减去孔数。下面介绍一种欧拉数的计数法。观察图像中所有 2×2 邻域。设 Q_v 为 2×2 邻域中仅有一个点为1的数目，Q_t 为有三个为1的邻域数，Q_d 为图中只有对角线两点为1的 2×2 邻域的数目。则可以证明[22,23]，全图的欧拉数为

$$E=\frac{1}{4}(Q_v-Q_t\pm 2Q_d) \tag{8.5-4}$$

当上式用于 S 为4连通，S^c 为8连通情况时，式中选用"+"号，反之则用"-"号。

8.5.2 面积

在数字图像中 S 的面积的简单度量就是 S 中像素的数目。因此，图像中不同部分 $S_1, S_2, \cdots, S_m$ 的面积计算就是直接统计各集合中点的总数。在二值图中物体的面积可以通过8.5.1小节中的标注算法来计算。由于每个物体各用不同的符号标注，所以只要在标注过程中记下各种符号的总数就可算得各个物体的面积。

从各种闭合边界或区域的描述计算面积也是方便的。例如，已知集合 S 的行程编码，只要把值为1的行程的长度相加即可。已知图像的四叉树表示，要求 S 或 S^c 的面积，只要将相应的叶节点的面积相加。若图像大小为 $N\times N$，$N=2^k$，从四叉树定义可知，叶节点的面积为 4^{k-h}，h 为这个叶节点所处的层数。如果给定了闭合边界的某种表示，那么相应的连通区域的面积应等于区域外部边界所包围的面积减去它的内部边界所包含的面积(孔的面积)。下面以8链码说明怎样从链码求相应边界所围的面积。

设起点坐标为(x_0, y_0)。第 k 段链终端的 y 坐标为

$$y_k=y_0+\sum_{i=1}^{k}\Delta y_i \tag{8.5-5}$$

式中：

$$\Delta y_i=\begin{cases}1, & \varepsilon_i=1,2,3\\ 0, & \varepsilon_i=0,4\\ -1, & \varepsilon_i=5,6,7\end{cases} \tag{8.5-6}$$

ε_i为第 i 个码元。设

$$\Delta x_i = \begin{cases} 1, & \varepsilon_i = 1, 0, 7 \\ 0, & \varepsilon_i = 2, 6 \\ -1, & \varepsilon_i = 3, 4, 5 \end{cases} \tag{8.5-7}$$

$$a = \begin{cases} 1/2, & \varepsilon_i = 1, 5 \\ 0, & \varepsilon_i = 0, 2, 4, 6 \\ -1/2, & \varepsilon_i = 3, 7 \end{cases} \tag{8.5-8}$$

则相应边界所围面积为：

$$A = \sum_{i=1}^{n} (y_{i-1}\Delta x_i + a) \tag{8.5-9}$$

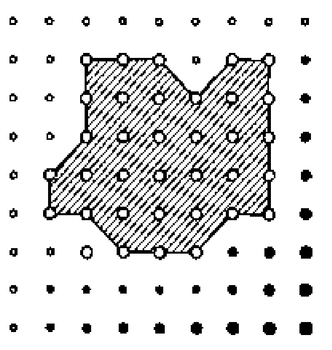




图 8.30　面积的一种定义(8 连通)

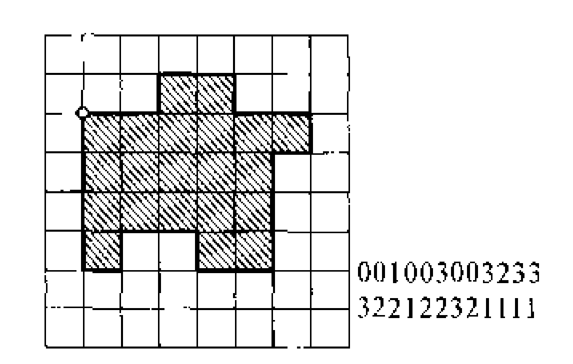


图 8.31　用"接缝"码求周长

式(8.5-9)所求面积是用 8 链码描述的边界内包含的单位方格数[24]，见图 8.30。若要用区域所含像素点的数目来表示面积，则要用所谓"接缝码"(Crack Code)来计算[11]。这是一种用 4 链码表示的边界码，如图 8.31 所示。此外，有定理(Picks Theorem[11])证明，若一个单连通区域的边界点数为 b，内部点数为 i，则这一区域的面积为

$$A = \frac{1}{2}b + i - 1 \tag{8.5-10}$$

8.5.3　周长及弧长

数字图像中的弧或曲线的长度可以从 8 链码得到。只要将水平和垂直链的长度记作 1，对角线方向的链记作$\sqrt{2}$，然后将它们相加即可。但是这种弧长的计算方法不适用于 4 链码，否则将会产生很大的误差。

至于区域的周长，按不同定义可以用不同的方法计算。一般用以下三种定义来近似：

(1) 若将图像中每个像素都看做单位面积的小方格，则区域 S 和背景 $\bar{S}$ 都由方格组成，区域 S 的周长可以定义成区域和背景交界线(接缝)的长度(图 8.31)。

(2) 将像素看做一个个的点，则周长可以定义为区域边界 8 链码的长度。

(3) 边界所占面积。

对图 8.31 所示区域，三种定义的周长分别为：

$$L = 24,\ L = 10 + 5\sqrt{2},\ L = 15$$

这三种计算方法都很简单。对第一种定义，只要计算接缝码的长度。这种码实际上是将区域与背景的"接缝"用普通 4 链码编码后所得结果。第二种定义的周长的计算方法与用 8 链码编码的曲线长度的计算方法相同。第三种定义的周长则为边界点数之和(小方格面积为 1)。

另外，从其他的区域描述也可以计算周长，例如行程编码[1]，四叉树表示[20]等。

第二种定义的周长，除了直接计算 8 链码的长度外，对于未经边界跟踪编码的阵列表示的图像，可用下述方法计算它的周长[26]。

设 S_i 为某一连通分量，P 为 S_i 中的一点。P 的 8 个邻点从右面一点开始按逆时针方向分别为 P_0，P_1，P_2，…，P_7。令

$$b_1(P) = \sum_{k=0}^{3}\left[(P \wedge P_{2k}) - (P_{2k} \wedge (P_{2k-1} \vee P_{2k-2}) \wedge (P_{2k+1} \vee P_{2k+2}))\right]$$

$$b_2(P) = \sum_{k=0}^{3}\left[(P \wedge P_{2k+1}) - (P_{2k} \wedge P_{2k+1} \wedge P_{2k+2})\right] \tag{8.5-11}$$

式中：P_i 的下标全作模 8 运算。P 和 P_i 的值为 0 或 1。则 S_i 的周长为：

$$L(S_i) = \frac{1}{2}\sum_{P \in S_i}\left[b_1(P) + b_2(P)\sqrt{2}\right] \tag{8.5-12}$$

关于数字图像中直线和曲线长度计算的精度可参见参考文献[27，28]。

离散曲线是连续曲线的一种近似。用以上各种方法计算的曲线长度与原来曲线相比存在着较大的误差，而且这个误差不会由于采样精度的提高而减小，从而限制了它们的使用范围。下面介绍一种利用积分几何中 Crofton 定理，从 Crofton 方向直径导出离散图像周长的计算方法。这种方法计算的误差较小，而且随着采样间隔的减小，计算的离散图像的周长越来越接近连续图像中物体的真实周长[28]。

设区域 S 周长为 L，$D(P, \theta)$ 表示直线 $x\cos\theta + y\sin\theta = P$ 与 S 截交的次数(图 8.32 中 l_1 与 S 截交 1 次，l_2 与 S 截交次数为 2)。

区域 S 的方向直径定义为：

$$D(\theta) = \int_0^{\infty} D(P, \theta)\mathrm{d}P + \int_0^{\infty} D(P, \theta + \pi)\mathrm{d}P \tag{8.5-13}$$

$D(\theta)$也就是 S 在 θ 方向的投影宽度。一般地，方向直径为广义方向投影的总和，在图 8.32 中，有

$$D(\theta_i) = (a_1 + a_2) + a_3 \quad (8.5-14)$$

则 S 的周长可定义为

$$L = \int_0^{\pi} D(\theta)\mathrm{d}\theta = \frac{1}{2}\int_0^{\infty}\int_0^{2\pi} n(P,\ \theta)\mathrm{d}P\mathrm{d}\theta \quad (8.5-15)$$

式中：$n(P,\ \theta)$为直线 $x\cos\theta + y\sin\theta = P$ 与 S 的边界截交的次数，因此有

$$n(P,\ \theta) = 2D(P,\ \theta) \quad (8.5-16)$$

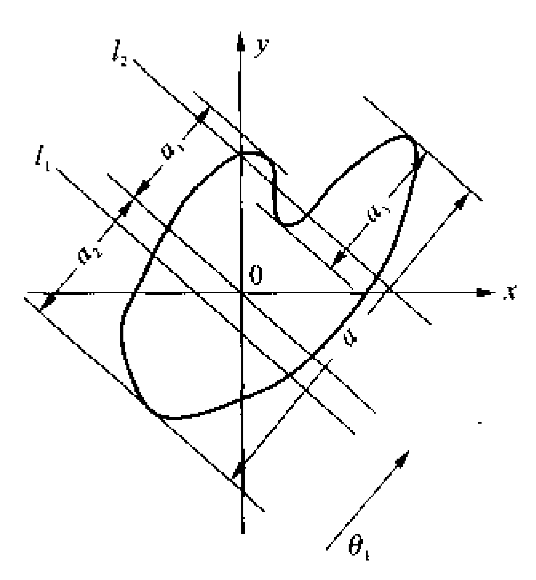


图 8.32 Crofton 方向直径

对离散图像，周长或任意闭合曲线的长度可用下式近似

$$L = \frac{\pi}{k}\sum_{i=0}^{k-1} n(\theta_i)\Delta \quad (8.5-17)$$

式中：$n(\theta_i)$为区域 S 与垂直于方向θ_i的直线族（等间距 Δ）截交的总次数，Δ 为直线族的间距。$n(\theta_i)\Delta = D(\theta_i)$ 为方向直径。k 为所取的方向数，通常取 4 或 8。

当方向数 $k = 8$ 时，$n(\theta_i)$可以用下式计算：

$$\begin{cases} n(0) = n\begin{pmatrix} 0 \\ 1 \end{pmatrix}, & n\left(\dfrac{\pi}{2}\right) = n(0 \quad 1) \\ n\left(\dfrac{\pi}{4}\right) = n\begin{pmatrix} 0 & \\ & 1 \end{pmatrix}/\sqrt{2}, & n\left(-\dfrac{\pi}{4}\right) = n\begin{pmatrix} & 1 \\ 0 & \end{pmatrix}/\sqrt{2} \end{cases} \quad (8.5-18)$$

$$\begin{cases} n\left(\dfrac{\pi}{2} + \arctan\dfrac{1}{2}\right) = \dfrac{1}{\sqrt{5}}\left[n\begin{pmatrix} \times & \times & 1 \\ 0 & \times & \times \end{pmatrix} + n\begin{pmatrix} \times & 1 & 0 \\ 0 & 1 & \times \end{pmatrix}\right] \\ n\left(\dfrac{\pi}{2} - \arctan\dfrac{1}{2}\right) = \dfrac{1}{\sqrt{5}}\left[n\begin{pmatrix} 0 & \times & \times \\ \times & \times & 1 \end{pmatrix} + n\begin{pmatrix} 0 & 1 & \times \\ \times & 1 & 0 \end{pmatrix}\right] \\ n\left(\dfrac{\pi}{2} + \arctan 2\right) = \dfrac{1}{\sqrt{5}}\left[n\begin{bmatrix} \times & 1 \\ \times & \times \\ 0 & \times \end{bmatrix} + n\begin{bmatrix} \times & 0 \\ 1 & 1 \\ 0 & \times \end{bmatrix}\right] \\ n\left(\dfrac{\pi}{2} - \arctan 2\right) = \dfrac{1}{\sqrt{5}}\left[n\begin{bmatrix} 0 & \times \\ \times & \times \\ \times & 1 \end{bmatrix} + n\begin{bmatrix} 0 & \times \\ 1 & 1 \\ \times & 0 \end{bmatrix}\right] \end{cases} \quad (8.5-19)$$

式中：$n(0,1)$，$n\begin{pmatrix} \times & \times & 1 \\ 0 & \times & \times \end{pmatrix}$等表示满足括号内取值的点阵的个数。×表示可取 0，也可取 1。

8.5.4 投影长度和直径

物体在 x 轴上的投影称为物体的宽度，在 y 轴上的投影称为高度。一个区域（物体）在 x 轴上的投影等于区域中最右点的 x 坐标和最左点的 x 坐标之差；y 轴上的投影则为区域中最高点的 y 坐标与最低点 y 坐标之差，见图 8.33。这两个特殊方向上的投影很容易从区域或区域边界的各种描述中求得。但求任意方向上的投影则比较复杂。下面以链码描述为例说明如何从链码求相应区域在 x 和 y 方向上的投影。

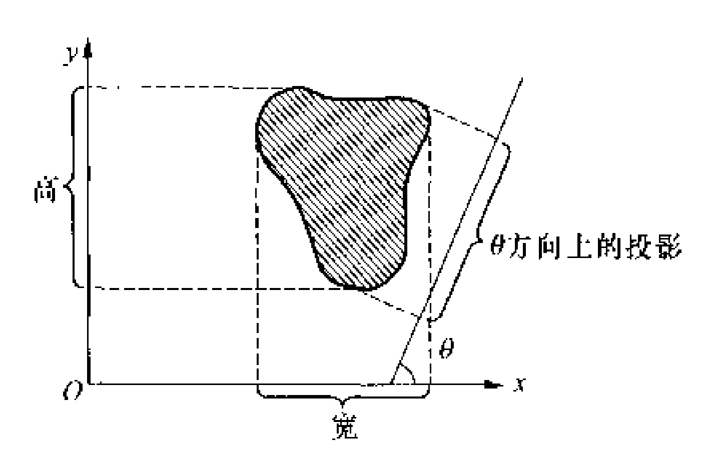


图 8.33 投影长度

设边界链码的起点坐标为(x_0, y_0)，边界点数为 n。设 x_{max}为最右点的 x 坐标，x_{min}为最左点的 x 坐标，第 k 点的 x 坐标为 x_k，并设

$$\Delta x_i = \begin{cases} 1, & \varepsilon_i = 0, 1, 7 \\ 0, & \varepsilon_i = 2, 6 \\ -1, & \varepsilon_i = 3, 4, 5 \end{cases}$$

ε_i为第 i 个码元。则求 x 轴上投影长度的算法如下：

(1) 置 $x_{max} = x_0$。

(2) 计算第 k 点的 x 坐标，$x_k = x_0 + \sum_{i=1}^{k} \Delta x_i$。

(3) 比较 x_{max}、x_{min}和 x_k的大小，并作如下处理：

$$x_{max} = x_k, \quad 若 x_{max} < x_k,$$

$$x_{min} = x_k, \quad 若 x_{min} > x_k。$$

(4) 从第 2 点开始($k=1$)重复以上步骤直至回到起点($k=n$)。

(5) 在 x 轴上的投影长度为 $P_x = x_{max} - x_{min} + 1$。

对于在任意方向 θ 上的投影,算法比较复杂。常用的方法是将 x 轴旋转相应的角度 θ 得 x',然后用上述方法求它在 x' 上的投影就是区域 S 在 θ 方向上的投影。

8.5.5 斜率和曲率

这里的斜率指曲线上某一点的切线的倾斜度,若用切线与 x 轴的夹角 θ 来表示,则在模拟情况下为:

$$\theta = \arctan \frac{dy}{dx} \tag{8.5-20}$$

在离散图像中边界某一点的导数常用边界点的差分来表示,即在 i 点的斜率为:

$$\theta_i = \arctan \frac{y_i - y_{i-1}}{x_i - x_{i-1}} \tag{8.5-21}$$

式中:x_i, y_i 为第 i 个边界点的 x, y 坐标。若边界用 8 链码表示,显然式(8.5-21)就是第 i 和第 $i-1$ 边界点之间指向符的角度。由此可见,斜率值都是 45°的整数倍。为了使斜率值的间隔更小、更加平滑,可以定义曲线上一点 P 前面 k 点或后面 k 点间的斜率作为 P 点的斜率。这种斜率可以用两种方法来近似:一是取前面 k 点斜率的平均值;另一种方法是连接两个端点,以连线的斜率作为 P 点的斜率。即将两端点坐标代入式(8.5-21)来计算。

斜率的变化率称为曲线的曲率,对离散图像仍用差分表示,即

$$C_i = \theta_i - \theta_{i-1} \tag{8.5-22}$$

对 8 链码来说,曲线上某点的曲率为这一点指向符所对应的角度与前一点指向符角度之差。因此,与斜率相似,任一点的曲率都是 45°的整数倍。为了使曲率变化更加平滑,同样也可定义某一点 P 的 k 点曲率,即取 P 点前后各 k 点计算 P 的曲率,但这时不宜采用各点曲率的平均值。至于 k 点斜率和 k 点曲率中 k 值怎样选取,应视具体应用而定,但是 k 与曲线或边界总长度相比应该很小。

对阵列表示的情况,可以取 P 点的适当大小的邻域来计算 P 点的斜率和曲率。显然,邻域的大小决定了斜率和曲率的平滑程度。具体方法在此不作介绍。

曲线的斜率和曲率在不少场合很有用。例如在曲线的搜索跟踪算法中,通常要计算已跟踪过的几点的斜率,以便提示下一步搜索的方向。

利用曲线的斜率和曲率可将曲线分割成斜率或曲率相近的几段,这种分

割的方法很多。其中一类方法类似于灰度分割，只是用曲线的斜率或曲线代替灰度。在讨论这些方法时认为只是简单的曲线或边界，即曲线或边界不出现交叉。

与灰度直方图类似，引入斜率直方图和曲率直方图，它们与灰度直方图有相同的含义。例如斜率直方图表明了每一个斜率值在曲线上发生的次数。对一个不太复杂的曲线，斜率直方图的峰值提供了曲线或边界总的方向信息。图 8.34 显示了一个闭合曲线、链码、斜率直方图和曲率直方图。曲率直方图反映了曲线的扭曲程度。对平滑的曲线，曲率在某个值附近，若曲线扭曲，则曲率分布较散。因此，可以利用曲率直方图确定适当的门限，将曲线分割成较直的部分和弯曲的部分。

另外，曲线或边界中的"角"是那些斜率有突变的点，也就是曲率的绝对值大的点。图 8.34(a)中以 x 为起点，用 * 表示曲率有局部极大值的点，可以用这些点将曲线分割成几部分。曲率的符号发生变化的点称为曲线的转折点，它们将曲线分为凸和凹两部分。

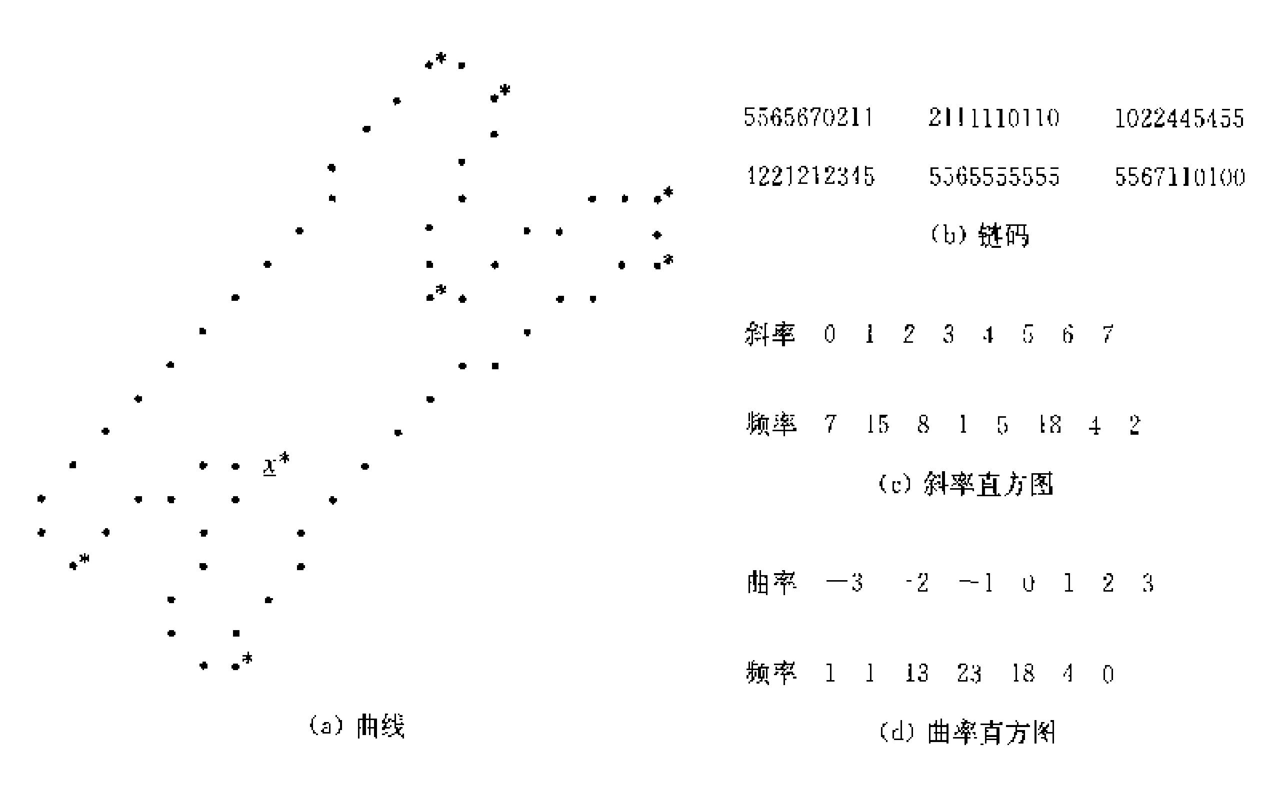


图 8.34 曲线的分割

8.5.6 形状特征的度量

这一小节讨论整体形状特征的度量。前述的矩、傅里叶描述子等也包含了整体的形状信息，这里从另一个角度加以讨论。

(1) 形状的复杂性。人们对形状复杂性的判断取决于好几个因素，其中包括

它的拓扑特性，例如孔洞数等。现在讨论的假设是单连通区域 S(即无孔存在)，因而只有一个闭合边界。

形状复杂性的一个度量是它的边界的扭曲程度(Wiggliness)。一种简单的计算方法是将整个边界的曲率的绝对值相加；另一种计算方法是计算曲率局部极大值的个数，并用它们的尖锐程度加权。

另一种形状复杂性的度量是 P^2/A，又称形状因子。其中 P 是区域的周长，A 是它的面积。周长采用平方是为了使这一度量与区域的大小无关。在一个实平面上，任何形状的复杂性都有

$$P^2/A \geqslant 4\pi \tag{8.5-23}$$

若物体为圆，则 $P^2/A = 4\pi$。若物体的形状伸长或不规则或边界扭曲，则 P^2/A 将增加。例如，正方形的 $P^2/A = 16$，正三角形为 $36/\sqrt{3}$等等。因此可以用 P^2/A 来度量形状的复杂性。

以上的数值是对连续图像而言的。对数字图像，由于周长计算存在较大的误差，使这一度量的误差也增大，在某种特定情况下甚至会失去其正确性。如果采用精度较高的周长测量方法，例如 8.5.3 小节最后所提出的周长计算法，则精度可以大大提高，而且随采样率的增加更接近于连续图像的情况。

复杂性还受到其他因素的影响。例如形状相同的部分、周期性、对称性等都使形状的复杂性下降。

(2) 伸长度(Elongatedness)。在 8.2.1 小节中讨论膨胀和收缩算法时，用 $S^{(k)}$ 表示 S 膨胀 k 次，$S^{(-k)}$ 表示 S 收缩 k 次。若 $S^{(-k)}$ 正好成空集，即收缩 k 次后 S 消失，则 S 中每一点离 S' 的距离都在 k 以内。令 $2k$ 为 S 的厚度，则 S 的伸长度可定义为：单连通区域 S 的伸长度为 A/t^2。A 是 S 的面积，t 是 S 的厚度。显然，只有当 A 比 k^2 大得多时才能说 S 的形状是伸长的。例如，$A \geqslant 10k^2$，即

$$A/t^2 \geqslant 10k^2/4k^2 = 2\times\frac{1}{2}$$

需要指出的是，伸长度反映了 S 的形状特征，但只有当 t 较大时才比较可靠，当 t 很小时已不足以说明问题。例如

$$\begin{matrix} 1 & 1 \\ 1 & 1 \end{matrix} \qquad \begin{matrix} 1 & 1 & 1 & 1 \end{matrix}$$

两种形状的面积都为 4，t 都为 2。但是只有第二种形状才能说是伸长的。

习　题

1. 设题图 8.1 中 S 为 8 连通，对它作收缩膨胀处理，以此说明收缩膨胀处理是不

可逆的，即 $(S^{(m)})^{(-m)} \neq S$。设 $m = 2$。

$$
\begin{matrix}
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\
0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 \\
0 & 0 & 1 & 1 & 1 & 1 & 1 & 0 \\
0 & 0 & 1 & 1 & 1 & 1 & 1 & 0 \\
0 & 1 & 1 & 1 & 1 & 1 & 1 & 0 \\
0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0
\end{matrix}
$$

题图 8.1

2. 在使用傅里叶描述子时，边界点坐标用复数 $z(l) = x(l) + jy(l)$ 表示。请写出它们的 DFT，说明什么情况下 DFT 仅有实数项(描述符中只有实数项)，这时边界形状有什么特点?

3. 证明三种距离有如下关系：

$$d_8(P, Q) \leqslant d_e(P, Q) \leqslant d_4(P, Q)$$

4. 试设计一迭代端点拟合法的算法。

5. 证明用 8 链码描述的曲线是闭合曲线的充分必要条件是：$n_0+n_1+n_7=n_3+n_4+n_5$，而且 $n_1+n_2+n_3=n_5+n_6+n_7$。

6. 证明如果 $f(x, y)$关于直线 $x=y$ 对称，即对所有 x 和 y 有 $f(x, y) = f(y, x)$，则此二维函数的$(i+j)$阶矩有 $m_{ij}=m_{ji}$，$\forall i, j$。

7. 证明任意多边形的面积为

$$A = \frac{1}{2}\sum_{j}(x_{j+1}y_j - x_j y_{j-1})$$

式中：j 为顶点序号；(x_j, y_j)为多边形顶点的坐标。连加从多边形任一顶点开始，沿逆时针方向顺序访问各个顶点，直至起始顶点。

8. 证明三次 B 样条函数有连续的一阶导数和连续的二阶导数。(提示：计算在相邻两区间连接处，即前一区间的终点和下一区间的起点，曲线的 B 样条函数及其导数的值，并比较之)。

9. 计算由链码 7700007656553554332 12321 所描述的闭合边界内部区域的面积。

10. 下面是一个物体的边界坐标，请算出它的形状因子：

X 2 3 4 5 6 7 7 7 7 7 6 5 4 3 2 1 1 2 2

Y 1 1 1 2 1 1 2 3 4 5 5 6 6 6 5 5 4 3 2

参考文献

[1] Rosenfeld A. Picture Languages: Formal Modals for Picture Recognition[M]. Academic Press, New York, 1979, Chapter 2.

[2] Nakagawa Y, Rosenfeld A. A note on the use of local min and max operations in digital picture processing[J]. IEEE Trans. Systems Man Cybernet, 8, 1978, 623~635.

[3] Rosenfeld A. A characterization of parallel thinning algorithm[J]. Informat. Control 29, 1975, 286~291.

[4] Dyer C R, Rosenfeld A. Thinning algorithms for gray scale pictures[J]. IEEE Trans. Pattern Anal. Maching Intelligence 1, 1979, 88~89.

[5] Yuan-hua Zhou, et al. Automatic description of the venation of mosquito wing from digitized images[J]. Syst. Zool., 31(3), 1985, 316~358.

[6] Deboor C A. Practical Guide to Splines[M]. New York: Springer Verlag, 1978.

[7] Freeman H. On the encoding of arbitrary geometric configurations[J]. IEEE Trans. Elec. Computers, Vol. EC-10, 1961, 260~268.

[8] Freeman H. Computer Processing of line drawing images[J]. Computer Survey 6, 1, March 1974, 57~98.

[9] Zahn C T, Roskies R Z. Fourier descriptors for plane closed curves[J]. IEEE Trans. On Computers, Vol. C-21, No. 3, March 1972, 269~281.

[10] Bennett J R, MacDonald J S. On the measurement of curvature in a quantized environment [J]. IEEE Trans. Vol. C-25, No. 8, August 1975, 803~820.

[11] Rosenfeld A, Kak A C. Digital Picture Processing, Second Edition[M]. Vol. 2, Academic Press, New York, 1982.

[12] Duda R O, Hart P E. Pattern Classification and Scene Analysis[M]. John Wiley & Sons, New York, 1973.

[13] Paoulis A. Probability, Random Variables, and Stochastic Process[M]. McGrawHill, New York, 1965.

[14] 冈萨雷斯,等. 数字图像处理(第二版)[M]. 阮秋琦,等译. 北京:电子工业出版社, 2003年.

[15] Hu M H. Visual pattern recognition by moment invariants[J]. IRE Trans. Info. Theory, Vol. IT-8, 1962, 179~187.

[16] Paton K. Conic Sections in chromosome analysis[J]. Pattern Recognition 2, 1970, 39~51.

[17] Samet H. Region representation: quadtrees from binary arrays[J]. CGIP 13, 1980, 88~93.

[18] Samet H. The quadtree and related hierarchical data structure[J]. ACM Computing Sur

vey. Vol. 16, 1984,187~260.

[19] Lauzon J P, Mark D M, et al. Two-dimensional run-encoding for quadtree representation [J]. CVGIP 30, 1985,36~69.

[20] Samet H. Region representation: Quadtrees from boundary codes[J]. Com. of the ACM, Vol. 23, No. 3, March 1980, 163~178.

[21] Dyer C R, Rosenfeld A, Samet H. Region representation: boundary codes from quadtrees [J]. Com. of the ACM, Vol. 23, No. 3, March 1980, 171~179.

[22] Gray S B. Local properties of binary images in two dimensions[J]. IEEE Trans. Comput. 20 1971,551~561.

[23] Minsky M L, Papert S. Perceptrons An Introduction to Computational Geometry[M]. MIT Press, Cambridge, Massachusetts, 1969.

[24] Ballard D H, Brown C M. Computer Vision[J]. Prentice Hall, Inc., New Jersy, 1982, 235~237.

[25] Hament H. Computing perimeters of images represented by quadtrees[J]. IEEE Trans, Vol. PAMI-3, 1981,638~687.

[26] Sanker P V, Krishnamurthy E V. On the compactness of subsets of digital pictures[J]. CGIP 8, 1978,136~143.

[27] Ellis T J, Proffit D, Rosen D, Rutkowski W. Measurement of the lengths of digitized curved lines[J]. CGIP 10, 1979,333~347.

[28] Proffit D, Rosen D. Metrication errors and coding efficiency of chain-encoding schemes for the representation of lines and edges[J]. CGIP 10, 1979,318~332.

[29] Castleman K R. 数字图像处理[M]. 朱志刚,等译,北京:电子工业出版社,1998.

[30] 吴敏金,万嘉若,王成道. Crofton 定理拓广与图像特征提取,数学的实践与认识[J]. 1983年第1期.

附录　数学形态学简介

附录 1　阈值化

设有图像 $f(x, y)$，其灰度级范围为$[z_0, z_K]$，T 是 z_0 和 z_K 之间的任意一个数。则设置阈值为 T 的结果可以得到一幅二值图像 $f_T(x, y)$，它由

$$f_T(x, y) = \begin{cases} 1, & f(x, y) \geqslant T \\ 0, & f(x, y) < T \end{cases} \tag{Ⅰ-1}$$

或者由

$$f_T(x, y) = \begin{cases} 0, & f(x, y) \geqslant T \\ 1, & f(x, y) < T \end{cases} \tag{Ⅰ-2}$$

定义。也可以选择双阈值，把所需灰度级范围内的像素变为 1，而把此范围外的像素变为 0，即

$$f_{(T_1, T_2)}(x, y) = \begin{cases} 1, & T_1 \leqslant f(x, y) \leqslant T_2 \\ 0, & \text{其他} \end{cases} \tag{Ⅰ-3}$$

或者

$$f_{(T_1, T_2)}(x, y) = \begin{cases} 0, & T_1 \leqslant f(x, y) \leqslant T_2 \\ 1, & \text{其他} \end{cases} \tag{Ⅰ-4}$$

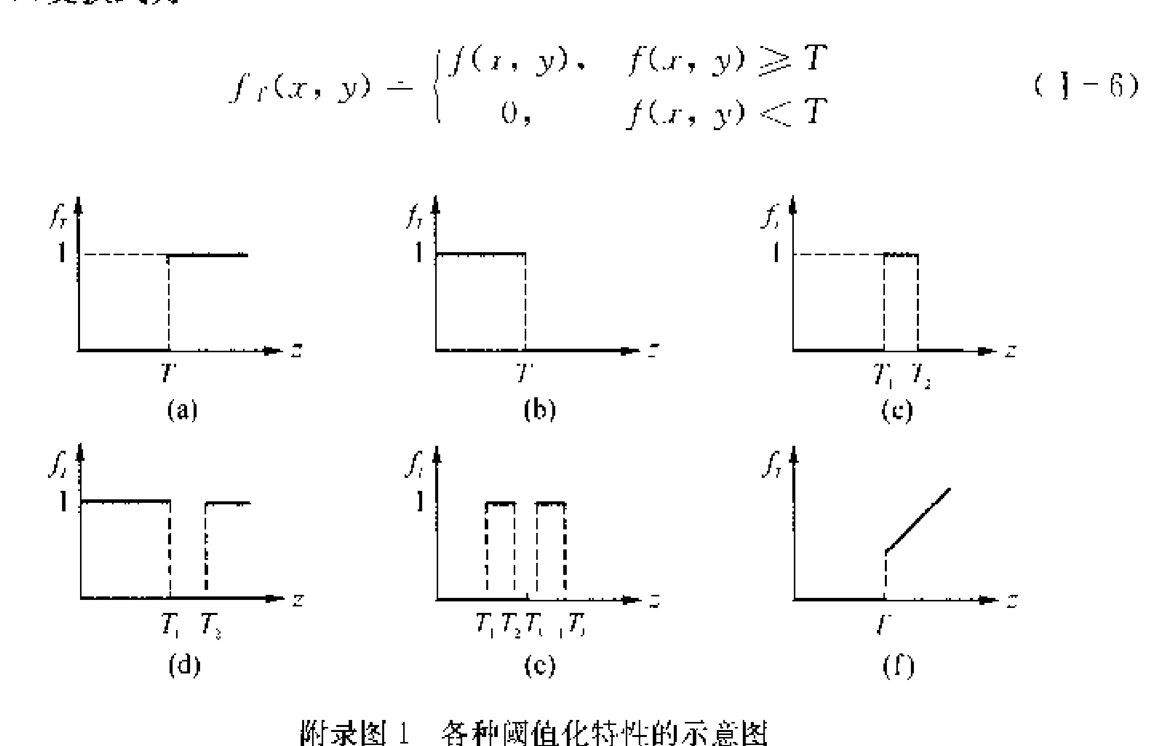
更为一般地，如果 Z 是一个任意的灰度级的集合，$Z \subseteq [z_0, z_K]$，就可以定义一种广义的“阈值化”运算，它把在 Z 中的灰度级变为 1，把不在 Z 中的灰度级变为 0，即

$$f_Z(x, y) = \begin{cases} 1, & f(x, y) \in Z \\ 0, & \text{其他} \end{cases} \tag{Ⅰ-5}$$

附录图 1 是上述各种阈值化特性的示意图。(a)～(e)分别与式(Ⅰ-1)～(Ⅰ-5)相对应。以上所述五种情况都是通过阈值化把多值图变为二值图，这里再介绍一

种称为“半阈值”的阈值化技术。它将某些灰度级(例如,低于阈值 T 的灰度级)变化为 0,而其余的灰度级保持不变。这样得到的仍是一个多值图,其特性参见附录图 1(f),变换式为

$$f_T(x, y) = \begin{cases} f(x, y), & f(x, y) \geqslant T \\ 0, & f(x, y) < T \end{cases} \tag{1-6}$$



附录图 1　各种阈值化特性的示意图

附录 2　形态滤波

近年来,在图像处理和机器视觉中越来越多地用到形态滤波。这和并行处理技术的迅速发展有着密切的关系。数学形态学运算本质上是并行的,便于用并行处理有效地实现。有些图像处理与分析系统把数学形态学运算作为其基本运算,由此出发来考虑其体系结构。以前介绍的不少边缘检测、边缘增强等方法是基于卷积运算的,这是一种空间域的线性滤波。而形态滤波的工作原理和性能与卷积有较大的不同。

本节介绍数学形态学基本运算的定义、性质及相互之间的关系。本附录内容取材于本附录的参考文献[1]、[2]、[3]。

附 2.1　二值图的形态滤波

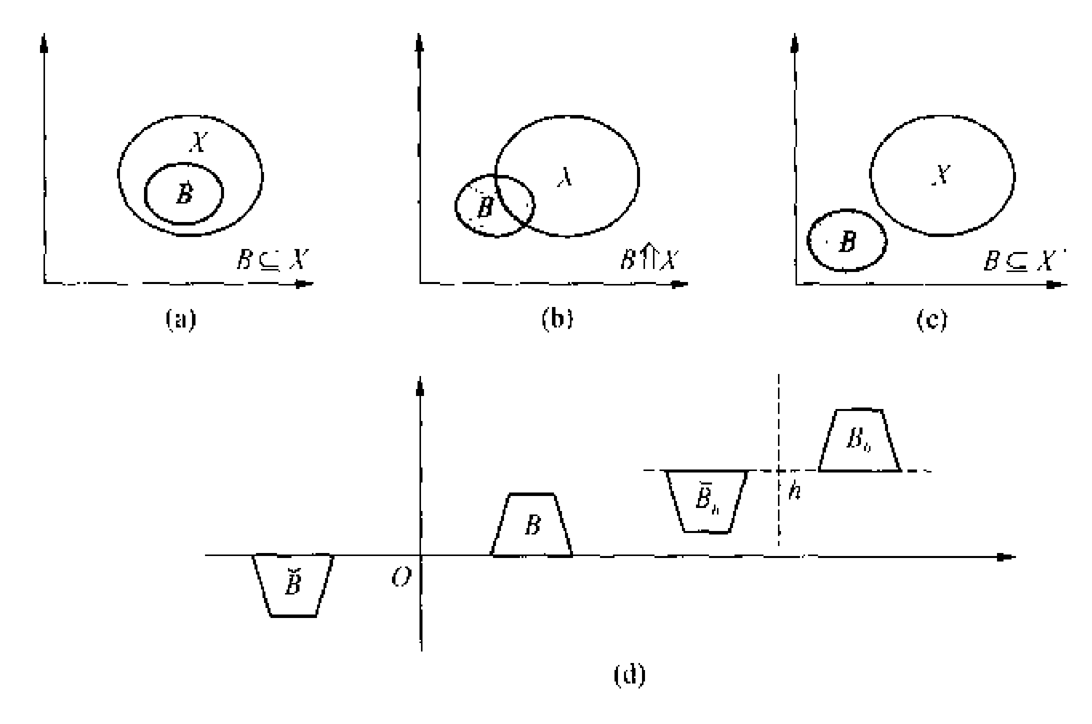
为了叙述的方便,现在先把讨论限制在二维欧氏空间的情况下,从基本符号、定义开始。设 X 和 B 是二维欧氏空间中的两个集合,它们各自对应于二值图中的不同点集。集合 X 的元素用 $x \in X$ 来表示,集合 B 的元素用 $b \in B$ 来表示。附录

图 2 示出 X 和 B 之间的三种关系：

(1) B 包含于 X，即 $B\subseteq X$；

(2) B 击中 X，记作 $B\Uparrow X$，即 $B\cap X\neq\varnothing$；

(3) B 击不中 X，记作 $B\subseteq X^c$，即 $B\cap X=\varnothing$。这里 X^c 为 X 的补集，$\varnothing$ 为空集。



附录图 2　X 和 B 之间的三种关系及平移对称示意图

在数学形态学运算中常把集合 B 称为结构元素，用 $\check{B}$ 表示 B 对于原点的对称集，即 $\check{B}=\bigcup_{b\in B}\{-b\}$，用 B_h 表示将 B 平移 h 所得到的集合。注意这里的 h 表示二维欧氏空间中的位移，若其横坐标和纵坐标分量分别为 h_x 和 h_y，则 B_h 表示将点集 B 沿横轴方向移 h_x，再沿纵轴方向移 h_y。请注意不要把 B_h 误解为只是在水平方向上移动 h，实际上 B_h 表示了二维平移。用 $\check{B}_h$ 表示将 B 先绕原点转 180°，然后平移 h。附录图 2 (d) 是 $\check{B}$、B_h 和 $\check{B}_h$ 含义的示意图。

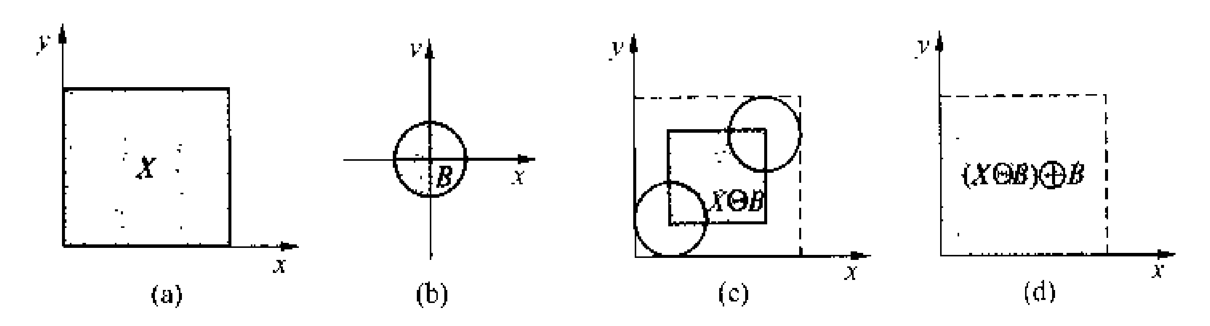
1. 腐蚀(Erosion)和膨胀(Dilation)

定义 1：腐蚀。

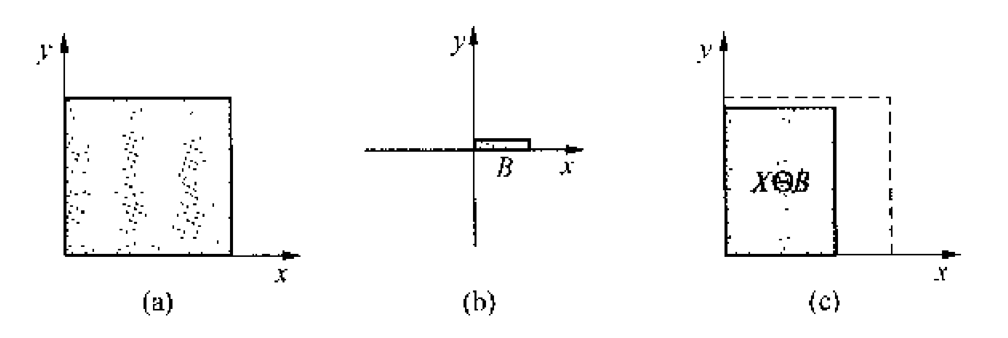
把结构元素 B 平移 h 以后得到 B_h，使 B_h 包含于 X 的所有 h 对应的点构成的集合称为 X 被 B 腐蚀的结果，用式子表示为：

$$X\Theta B=\{h\mid B_h\subseteq X\}\qquad(\text{II}-1)$$

附录图 3 和附录图 4 分别给出了 B 对称情况下和 B 不对称情况下腐蚀的例子。



附录图3　对称 B 情况下的腐蚀运算



附录图4　B 不对称情况下的腐蚀运算

上面所说的腐蚀的定义是从平移集合 B 来说明的，事实上也可以从平移 X 来说明，即 $X\Theta B=\bigcap_{b\in B}X_{-b}=\bigcap_{b\in \breve{B}}X_b$，请读者自己证明该关系成立。

附录图3形象地说明了把 $X\Theta B$ 称为腐蚀的原因。但是，认识不能停留在感性上。若 B 是不为原点的单点集，即 $B=\{a\}$，则由上式可知 $X\Theta B=X_{-a}$，这样，腐蚀运算只相当于平移而没有"腐蚀"掉任何东西。

定义2：膨胀。

使 $\breve{B}_h$ 击中 X 的所有 h 对应的点构成的集合称为 X 被 B 膨胀，记作

$$X\oplus B=\{h\mid \breve{B}_h\cap X\neq\varnothing\}\tag{Ⅱ-2}$$

虽然可以像讨论腐蚀时一样推出与膨胀运算有关的公式，但可以用对偶原理来简化公式的推导。先从一个日常生活的例子来看对偶性。有人观察河岸，发现河岸被腐蚀了。这同样的现象，如果由观察河水的人来报告，则结果是河面膨胀了。河岸和河面构成两个互补的集合。同样的一个过程，若从集合 X 的角度看，X 受到 B 的腐蚀。那么从另一个角度，即从集合 X 的补集 X^c 的角度看，X^c 得到了膨胀，用式子表示就是

$$(X\Theta B)^c=X^c\oplus\breve{B}\tag{Ⅱ-3}$$

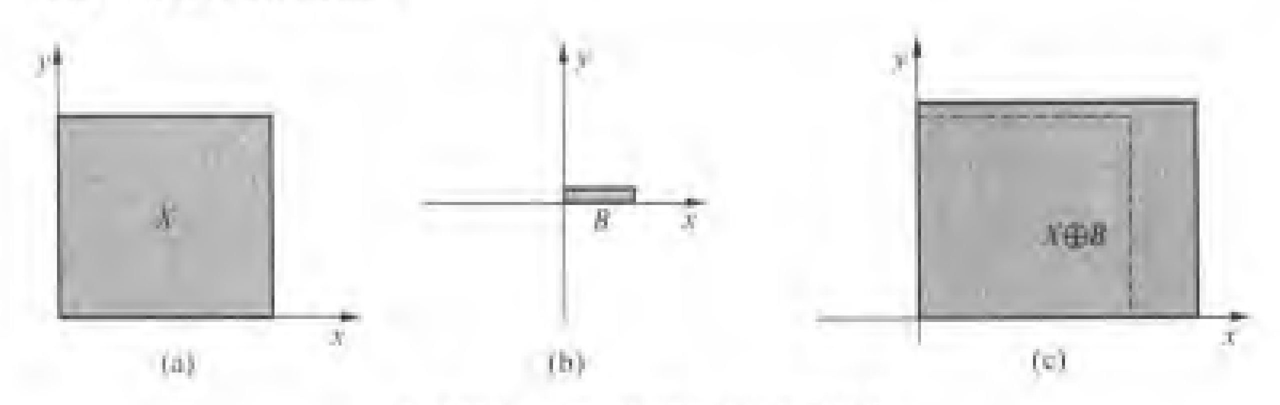
由上述对偶性的原理，等式两边应该相等。此式也可写成

$$(X\Theta\check{B})^c = X^c \oplus B \tag{Ⅱ-4}$$

于是，由式（Ⅱ-3）和式（Ⅱ-4）容易推得

$$X \oplus B = (X^c\Theta\check{B})^c = \left[\bigcap_{b\in\check{B}} X^c_{-b}\right]^c = \bigcup_{b\in B} X_b \tag{Ⅱ-5}$$

在下面的讨论中，我们还将经常用对偶性来简化讨论。附录图 5 是 B 不对称情况下 $X\oplus B$ 的示意图。



附录图 5　B 不对称情况下的膨胀运算

$X\oplus B$ 和 $X\Theta B$ 又称为结构和及结构差运算。下面讨论它们的一些性质：

（1）$X \oplus B = B \oplus X$，但 $X\Theta B \neq B\Theta X$。

（2）若 $X\subseteq X'$，则

$$X\Theta B \subseteq X'\Theta B，\forall B \tag{Ⅱ-6a}$$

$$X \oplus B \subseteq X' \oplus B，\forall B \tag{Ⅱ-6b}$$

若 $B\subseteq B'$，则

$$X\Theta B \supseteq X\Theta B'，\forall X \tag{Ⅱ-6c}$$

也就是说，结构和有单调增加性；结构差对 X 有单调增加性，而对 B 则是单调减少的。

（3）若 B 为单点集 $\langle b\rangle$，则以它为结构元素作结构和或结构差相当于对 X 作平移，即

$$X \oplus \langle b\rangle = X\Theta\langle b\rangle = X_b \tag{Ⅱ-7a}$$

容易进一步推得

$$X_b \oplus B = (X \oplus B)_b \tag{Ⅱ-7b}$$

(4) 由性质(3)立即可得 $X \oplus \{0\} = X \ominus \{0\} = X$。再进一步,容易证明:

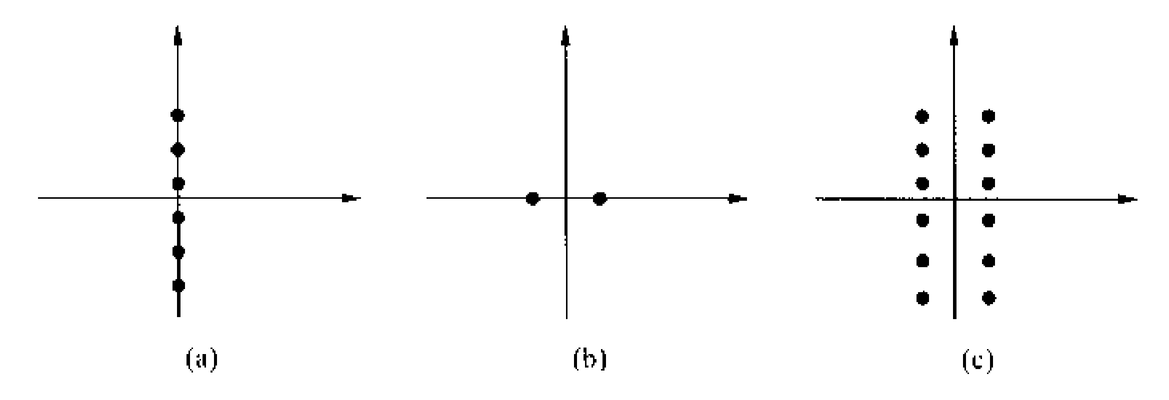
若 $0 \in B$,则 $X \subseteq X \oplus B$,以及 $X \ominus B \subseteq X$;

若 $0 \notin B$,则不一定有 $X \subseteq X \oplus B$ 成立。附录图 6 给出了这样的例子。由于 $0 \notin B$,所以 $X \oplus B$ 由左右两行点构成,却不包括原来位置上的一排点。

(5) 分配律,即

$$X \oplus (B \cup B') = (X \oplus B) \cup (X \oplus B') \tag{Ⅱ-8a}$$

$$X \ominus (B \cup B') = (X \ominus B) \cap (X \ominus B') \tag{Ⅱ-8b}$$



附录图 6 当 B 不包括原点时 $X \oplus B$ 不包括 X 的例子

$$(X \cap Z) \ominus B = (X \ominus B) \cap (Z \ominus B) \tag{Ⅱ-8c}$$

证:(只证第一个等式,其余的容易从对偶性得到证明)

$$X \oplus (B \cup B') = (B \cup B') \oplus X = \bigcup_{x \in X} (B \cup B')_x$$

$$= \bigcup_{x \in X} (B_x \cup B'_x) = \left(\bigcup_{x \in X} B_x \right) \cup \left(\bigcup_{x \in X} B'_x \right)$$

$$= (B \oplus X) \cup (B' \oplus X) = (X \oplus B) \cup (X \oplus B')$$

式(Ⅱ-8a)和(Ⅱ-8b)表明,在做膨胀(或腐蚀)运算时可以分块进行,然后将分块进行膨胀(或腐蚀)的结果求并(或求交)就得到所要的结果。在式(Ⅱ-8c)中集合 Z 常代表观测窗的形状和大小。如果只能观察到在 Z 窗内的 X,即 $X \cap Z$,则式(Ⅱ-8c)表明求 $X \ominus B$ 操作的结果只是在 $Z \ominus B$ 区域内才是正确的。

(6) $X \oplus (B \cap B') \subseteq (X \oplus B) \cap (X \oplus B')$。 (Ⅱ-9a)

注意式中用的是包含号而不是等号。在什么情况下两边会不相等呢?现举一例说明。在附录图 7 中 B 和 B' 为过原点的左右两圆,则 $B \cap B' = \{0\}$,即交集为单点集,只有原点 O 一个像素。由性质(4),等式左边等于 X。而 $X \oplus B$ 与 $X \oplus B'$ 如附录图 7 中的(c)、(d)所示,其交集如附录图 7(e)所示。它比 X 大,大的部分在图

中用阴影画出。可见式(Ⅱ-9a)成立。

由对偶性可知：

$$X\Theta(B\cap B')\supseteq(X\Theta B)\cup(X\Theta B') \quad (Ⅱ-9b)$$

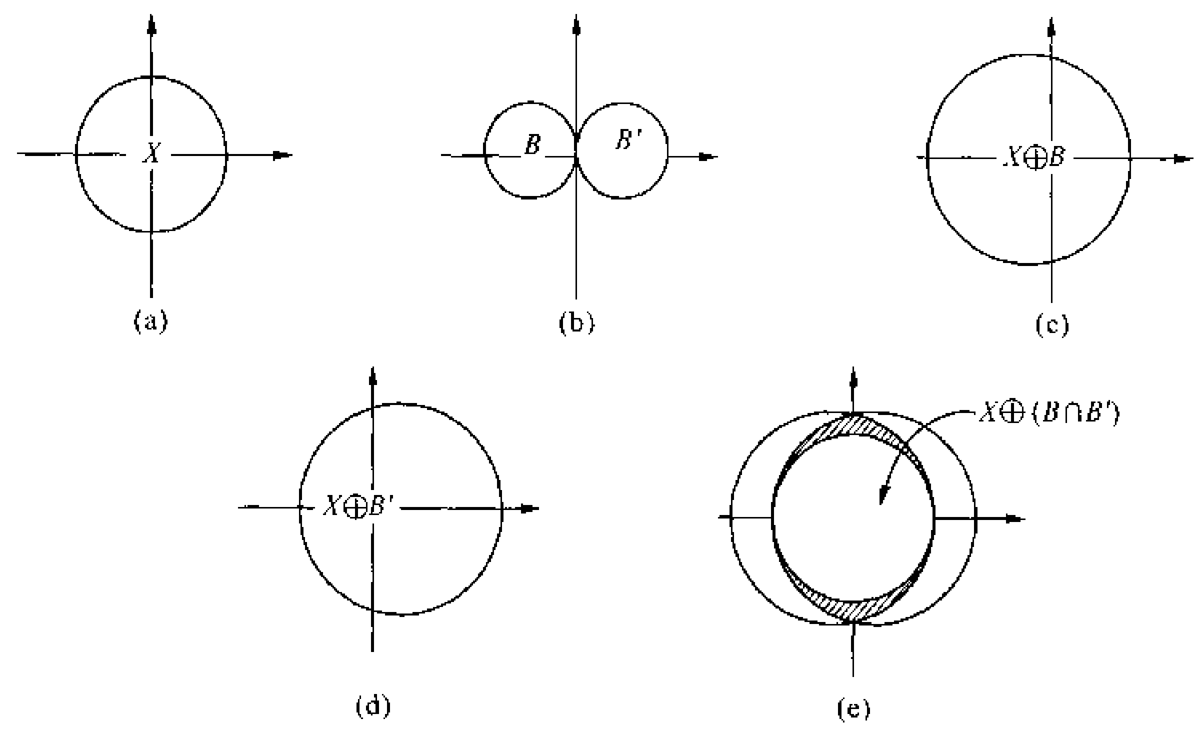
$$(B\cup B')\Theta X\supseteq(B\Theta X)\cap(B'\Theta X) \quad (Ⅱ-9c)$$

(7) 结构运算的多次迭代：

$$(X\Theta B)\Theta B' = X\Theta(B\oplus B') \quad (Ⅱ-10a)$$

$$(X\oplus B)\oplus B' = X\oplus(B\oplus B') \quad (Ⅱ-10b)$$

若原来要进行结构差运算的结构元素为正方形，可以把它看成 $B\oplus B'$，其中 B 为水平线段，B' 为垂直线段。则由式(Ⅱ-10a)，X 对 $B\oplus B'$ 作结构差相当于先对水平线段作结构差运算，然后再对垂直线段 B' 再作一次结构差。若再推广一下，用 X 对水平、60度、120度三个方向的等长线段连做三次结构差，就相当于 X 对正六边形的结构元素做结构差运算。可见，本性质在实际应用中是很有用的。



附录图 7 说明 $X\oplus(B\cap B')\subseteq(X\oplus B)\cap(X\oplus B')$ 的示意图

2. 结构开(Open) 和结构闭(Close)

一般来说，X 在经 B 腐蚀后不能靠再用 B 进行膨胀来完全恢复原始 X。附录图 3(d)示出这种情况，方形的 X 先被对称的圆形 B 腐蚀得到的仍是小方形，但将

$X\Theta B$ 再膨胀得到的$(X\Theta B)\oplus B$ 是磨去了四角的方形，即这样丢失了原来 X 中的若干细节，无法恢复原来形状 X。

定义 3：结构开。

$$X \circ B = (X\Theta B) \oplus B \qquad (\text{Ⅱ}-11)$$

式中：∘是结构开的符号，即先腐蚀后膨胀的运算。

定义 4：结构闭。

$$X \bullet B = (X \oplus B)\Theta B \qquad (\text{Ⅱ}-12)$$

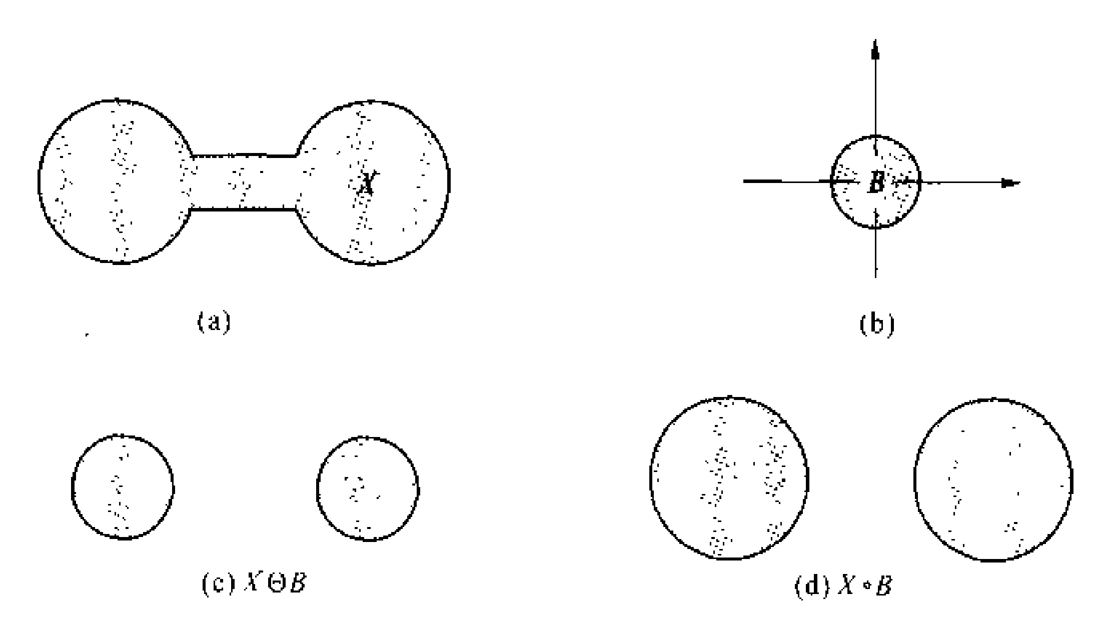
由对偶式容易证明

$$X^c \circ \check{B} = (X \bullet B)^c \qquad (\text{Ⅱ}-13\text{a})$$

$$(X \circ B)^c = (X^c) \bullet \check{B} \qquad (\text{Ⅱ}-13\text{b})$$

证：Ⅱ-13a 式右边 $=((X\oplus B)\Theta B)^c=(X\oplus B)^c\oplus\check{B}=(X^c\Theta B)\oplus\check{B}=(X^c)\circ\check{B}$

附录图 8 示出结构开运算的一例，它形象地说明了结构开名字的含义。左右两个本来相连的岛屿经 $X\circ B$ 运算后分开了。如果换一个观点，从 X^c 来看这个问题，经过运算，上下两个本来不通的海域现在相通了（闭）。一般说来，结构开去除孤立的小点、毛刺和小桥；而结构闭则填平小湖，弥合小裂缝。



附录图 8　结构开运算的例子

结构开与结构闭运算的性质：

(1) 结构开有非扩张性，结构闭有扩张性，即

$$X \circ B \subseteq X \subset X \bullet B \qquad (\text{Ⅱ}-14)$$

(2) 单调增加性

$$若\ X \subseteq X'，则\ X \circ B \subseteq X' \circ B\ 以及\ X \bullet B \subseteq X' \bullet B \qquad (Ⅱ-15)$$

(3) 等幂性，即

$$(X \circ B) \circ B = X \circ B，以及(X \bullet B) \bullet B = X \bullet B$$

证：由非扩张性知 $(X \circ B) \circ B \subseteq X \circ B$，而

$$\begin{aligned}(X \circ B) \circ B &= (((X \Theta B) \oplus B) \Theta B) \oplus B \\ &= ((X \Theta B) \bullet B) \oplus B \supseteq (X \Theta B) \oplus B = X \circ B\end{aligned}$$

故只能 $(X \circ B) \circ B = X \circ B$。同理可证$(X \bullet B) \bullet B = X \bullet B$。

从概念上想，$X \circ B$ 已经把 X 的毛刺等细节去掉了，再做一次对 B 的结构开，没有更多的毛刺可消去，因此，第二次结构开运算不再改变其形状。因此有等幂性。在实际做 $X \circ B$ 运算时会发现，对某些 X 有 $X \circ B$ 小于 X，而有时 $X \circ B = X$。可以证明，若存在一个集合 D，并且 $X = D \oplus B$，即若 X 是由某个集合 D 对 B 作结构和运算生成的，则 $X \circ B = X$。

3. 细化与粗化

为方便说明，先介绍击中击不中变换的含义。设想用两支捆在一起的箭去射击目标，并要求其中一支完全击中目标而另一支击不中目标。当箭的位置处在目标边界上时会出现这种情况，如附录图 9(a)。把所有符合这种情况的箭的位置点 p 构成的集合称为击中击不中变换的结果。击中击不中变换可用来寻找具有特定形状性质的像素集合(如边界)。

定义 5：击中击不中变换(HMT)。

设存在两个不相交集合 B_1 和 B_2，令 $B = \{B_1, B_2\}$，即 B 由两个集合 B_1 和 B_2 所组成，B 又称为复合结构元素。这里左边的集合 B_1 是要求击中的部分，右边的 B_2 是要求击不中的部分，左右两个集合不能随便颠倒次序。

$$X \otimes B = \{h \mid B_{1h} \subseteq X,\ B_{2h} \subseteq X^c\} \qquad (Ⅱ-16)$$

附录图 9(b)、(c)、(d)和(e)、(f)分别给出了两种情况下的击中击不中变换。图中 $X \otimes B$ 相应于 X 的右边，而 $X \otimes B'$ 相应于右边的外部。若 $B_2 = \varnothing$，则式(Ⅱ-16)退化成式(Ⅱ-1)，即 $B_2 = \varnothing$ 的 HMT 相当于腐蚀运算 $X \Theta B_1$。对式(Ⅱ-16)可作进一步演化如下：

$$\begin{aligned}X \otimes B &= \{h \mid B_{1h} \subseteq X\} \cap \{h \mid B_{2h} \subset X^c\} \\ &= (X \Theta B_1) \cap (X^c \Theta B_2) = (X \Theta B_1) \cap (X \oplus \check{B}_2)^c\end{aligned}$$

故
$$X \otimes B = (X \ominus B_1) \backslash (X \oplus \breve{B}_2) \qquad (\text{Ⅱ}-17)$$

式中：\ 是两个集合之差的运算符。

定义 6：细化。

细化的定义如下：

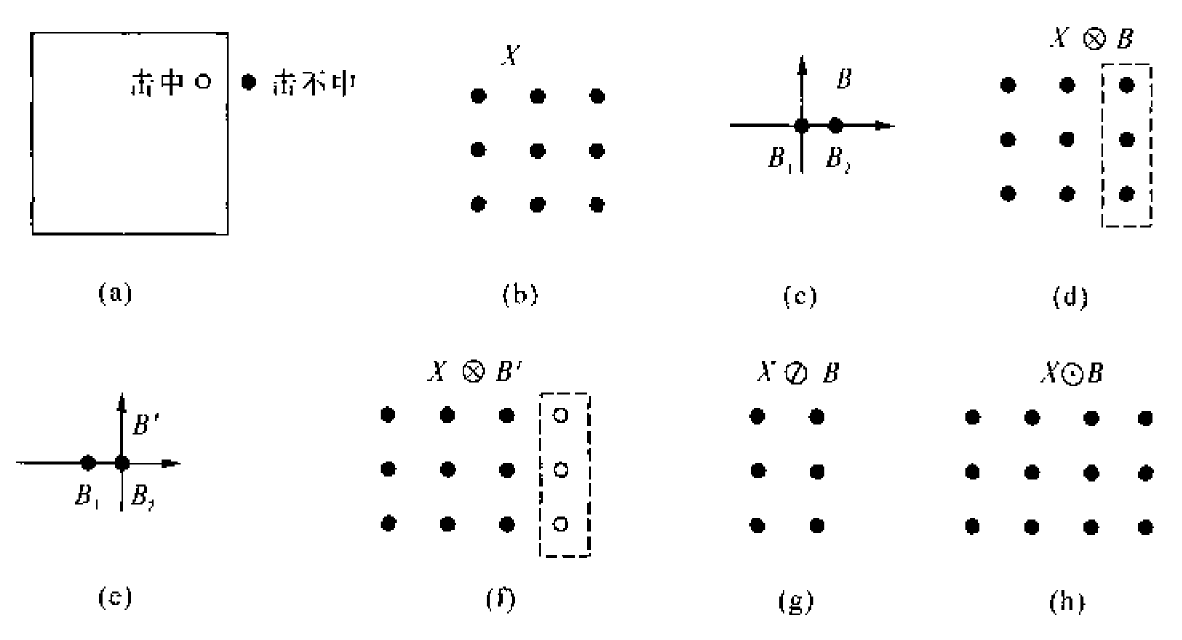
$$X \oslash B = X \backslash (X \otimes B) \qquad (\text{Ⅱ}-18)$$

注意式中 B 为复合结构元素。由上所述可得

$$X \oslash B = X \cap ((X \ominus B_1) \cap (X^c \ominus B_2))^c \qquad (\text{Ⅱ}-19)$$

定义 7：粗化。

$$X \odot B = X \cup (X \otimes B) \qquad (\text{Ⅱ}-20)$$



附录图 9　击中击不中及细化粗化的例子

附录图 9(g)和(h)分别给出了细化和粗化的例子，它们形象地解释了细化与粗化这两个名词的含义。由对偶性容易想到，在对 X 作细化的同时，就是对 X^c 作粗化。二者之间有下列关系式

$$(X \odot B)^c = X^c \oslash B^* \qquad (\text{Ⅱ}-21)$$

式中：$B^* = (B_2, B_1)$。

证：$(X \odot B)^c = \big(X \cup (X \otimes B)\big)^c = X^c \cap (X \otimes B)^c = X^c \backslash (X \otimes B)$

$$= X^c \backslash ((X \ominus B_1) \cap (X^c \ominus B_2)) = X^c \backslash ((X^c \ominus B_2) \cap ((X^c)^c \ominus B_1))$$

$$= X^c \backslash ((X^c \ominus B_2) \cap (X^c \oplus \breve{B}_1)^c) = X^c \backslash ((X^c \ominus B_2) \backslash (X^c \oplus \breve{B}_1))$$

$$= X^c \backslash (X^c \otimes B^*) = X^c \oslash B^*$$

关于粗化和细化的性质，我们不加证明地列出两条：

(1) 若 $o \in B_1$，则对所有 B_2有 $X \odot B = X$；

若 $o \in B_2$，则对所有 B_1有 $X \oslash B = X$；

读者可以对附录图 9 中所示情况求出 $X \odot B$ 以及 $X \oslash B'$，即可形象地验证上述结论。

(2) 如果 $B'_1 \subseteq B_1$，以及 $B'_2 \subseteq B_2$，我们把它记作 $B' \subseteq B$。则由 $B' \subseteq B$ 可以得到：

$$X \oslash B' \subseteq X \oslash B \subseteq X \subseteq X \odot B \subseteq X \odot B' \qquad (\text{Ⅱ}-22)$$

最后我们给出条件粗化和条件细化的定义。

定义 8：条件粗化。

条件粗化就是在 X 内对 Y 进行粗化，用式子来表示就是

$$Z = (Y \odot B) \cap X \qquad (\text{Ⅱ}-23)$$

$Y \odot B$ 是个集合，它的一部分元素属于集合 X，另一部分则不属于集合 X，用交运算把 $Y \odot B$ 中属于 X 的部分选出，这就是条件粗化的结果。举个形象一点的例子，把充了气的气球放入瓶内，然后加温使气球膨胀。那么，气球再膨胀也不能穿出瓶壁，这就是条件粗化，本例中原来的球是 Y，瓶子是 X，在瓶内胀大了的气球就是条件粗化的结果 Z。

定义 9：条件细化。

条件细化就是在细化 Y 时，使细化结果仍然包含 X，用式子表示就是

$$Z = (Y \oslash B) \cup X \qquad (\text{Ⅱ}-24)$$

举个形象点的例子，把梅子风干，就会缩小，但再小不会比梅核更小。这里，新鲜梅子相当于 Y，核相当于 X，干梅子相当于 Z。

附 2.2 灰度图像的形态滤波

灰度图像 $f(x, y)$可以看成一个函数。它相应于三维欧氏空间中的点集$\{(x, y, f(x, y))\}$，这里增广的第三维 z 相应于图像的灰度值。本影(umbra)是灰度图像形态滤波中的一个重要概念。设想在 z 轴正方向无穷远处有光源，则本影相应于被 $f(x, y)$遮住的阴影空间。它在负 z 方向伸展到无穷远。

定义 10：本影。

$$U(f) = \{p = (x, y, z);\ z \leqslant f(x, y)\} \qquad (\text{Ⅱ}-25)$$

而函数 f 是 $U(f)$ 的表面，即朝向 z 的正方向的曲面，不包括侧面，即

$$f = S(U(f)) \quad (\text{Ⅱ}-26)$$

采用 $U(f)$ 表示方法的好处在于对 $U(f)$ 进行并、交、腐蚀和膨胀等运算的结果仍然是一个本影。容易想象：

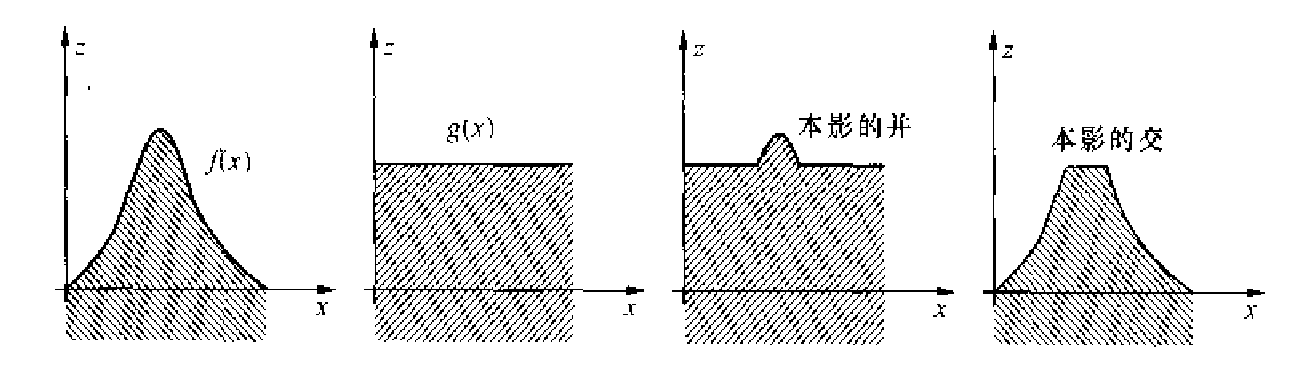
$$U(f) \cup U(g) = U(\max(f, g)) \quad (\text{Ⅱ}-27)$$

$$U(f) \cap U(g) = U(\min(f, g)) \quad (\text{Ⅱ}-28)$$

$U(f) \subseteq U(g)$ 意即

$$f(x, y) \leqslant g(x, y), \quad \forall (x, y) \in R^2 \quad (\text{Ⅱ}-29)$$

附录图 10 是上面三式的示意图。为了作图的方便，只画出二维空间，x 相应于空间位置，z 对应于灰度。由式(Ⅱ-29)看出，显然图中 $U(f)$ 和 $U(g)$ 之间没有相互包含的关系。

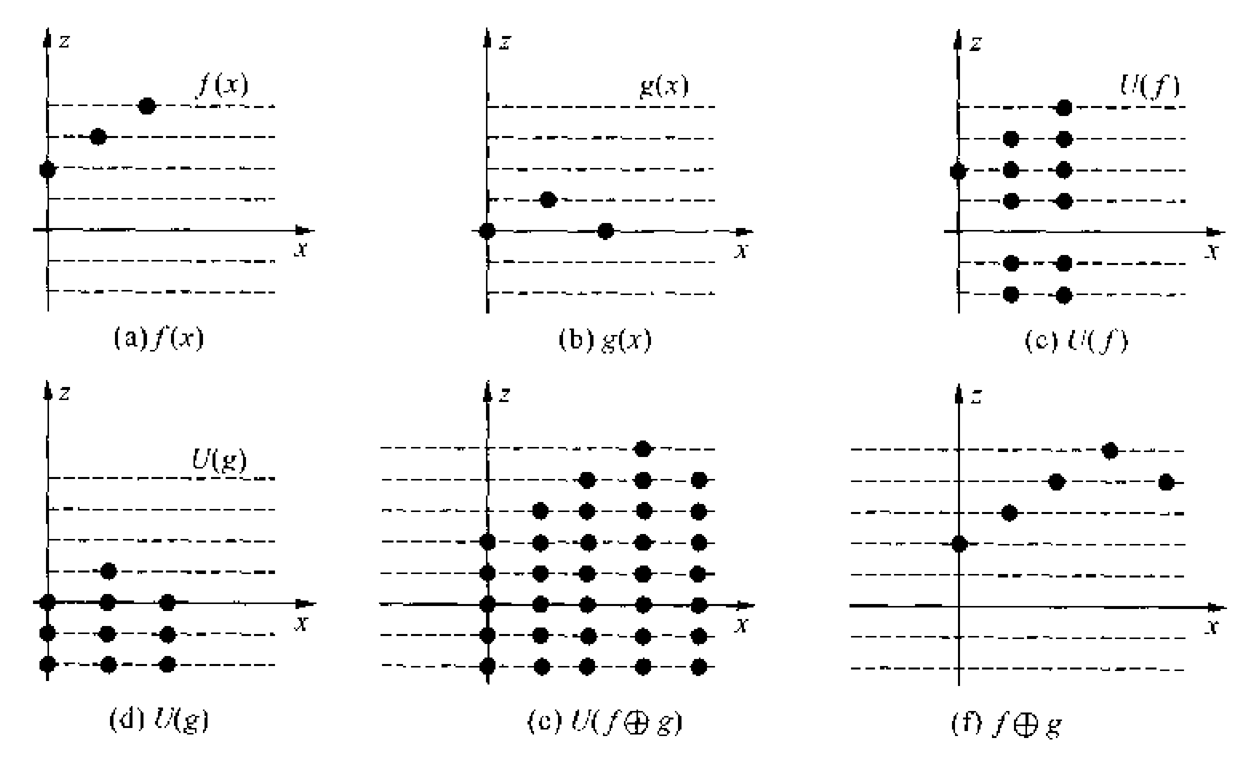


附录图 10　本影的并、交运算

利用本影可以定义两个函数 f 和 g 的结构和。由于本影是在增广了的欧氏空间中的集合，所以 $U(f)$ 和 $U(g)$ 之结构和是有意义的，它仍然是一个本影。这个本影所对应的函数就是函数 f 和 g 的结构和，即

$$U(f \oplus g) = U(f) \oplus U(g) \quad (\text{Ⅱ}-30)$$

上式右边的结构和是对增广了灰度这一维之后函数 f 和 g 的本影来进行的，其含义就是在三维空间中普通集合的结构和，在上小节的式(Ⅱ-5)中定义过。附录图 11 是在离散情况下，由 f 和 g 产生的 $f \oplus g$ 的示意图。这里为了作图的方便只画出一维情况。为了考虑灰度值的影响，将一维增广成两维，z 表示灰度值。在了解一维情况下灰度"图像"的结构和的形成过程以后，不难理解二维情况。由式(Ⅱ-30)，两个灰度图像结构和的本影就是 $U(f) \oplus U(g)$。在二维情况下，增广了灰度维之后，本影 $U(f)$ 及 $U(g)$ 都是三维空间中的点集。由式(Ⅱ-5)，求 $U(f) \oplus U(g)$ 时需要将 $U(f)$ 在三维空间中作移动并且求并集。这种移动不仅在 x 和 y 方向上，



附录图 11　离散灰度"图像"的结构和(一维情况)

也要在 z 方向上进行。在 z 方向上移动的大小与 $g(x, y)$ 值相对应。由式(Ⅱ-27),求两个本影的并集与取大(max)运算相对应。综合考虑上述的三维平移和取大运算之后,可以得到两个灰度图像 f 和 g 的结构和的表达式,即

$$(f \oplus g)(x, y) = d(x, y) = \max_{(i,j)}[f(x-i, y-j) + g(i, j)] \qquad (Ⅱ-31)$$

等式左边 $(f \oplus g)(x, y)$ 表示 f 和 g 结构和的结果图像在 (x, y) 点上的灰度值。等式右边的 (x, y) 和 (i, j) 表示点的坐标,取大运算是对使 $g(i, j)$ 有定义,也就是 $g(i, j)$ 不为负无穷的所有点 (i, j) 来进行的。而 f 和 g 的卷积可表示为:

$$(f * g)(x, y) = \sum_{(i,j)} f(x-i, y-j) \cdot g(i, j) \qquad (Ⅱ-32)$$

两式相比,在卷积中的累加 $\sum$ 在式(Ⅱ-31)中改成了取大运算 max,卷积中的乘号在式(Ⅱ-31)中改成了加号。完全类似,可以定义灰度图像的结构差,用本影表示有

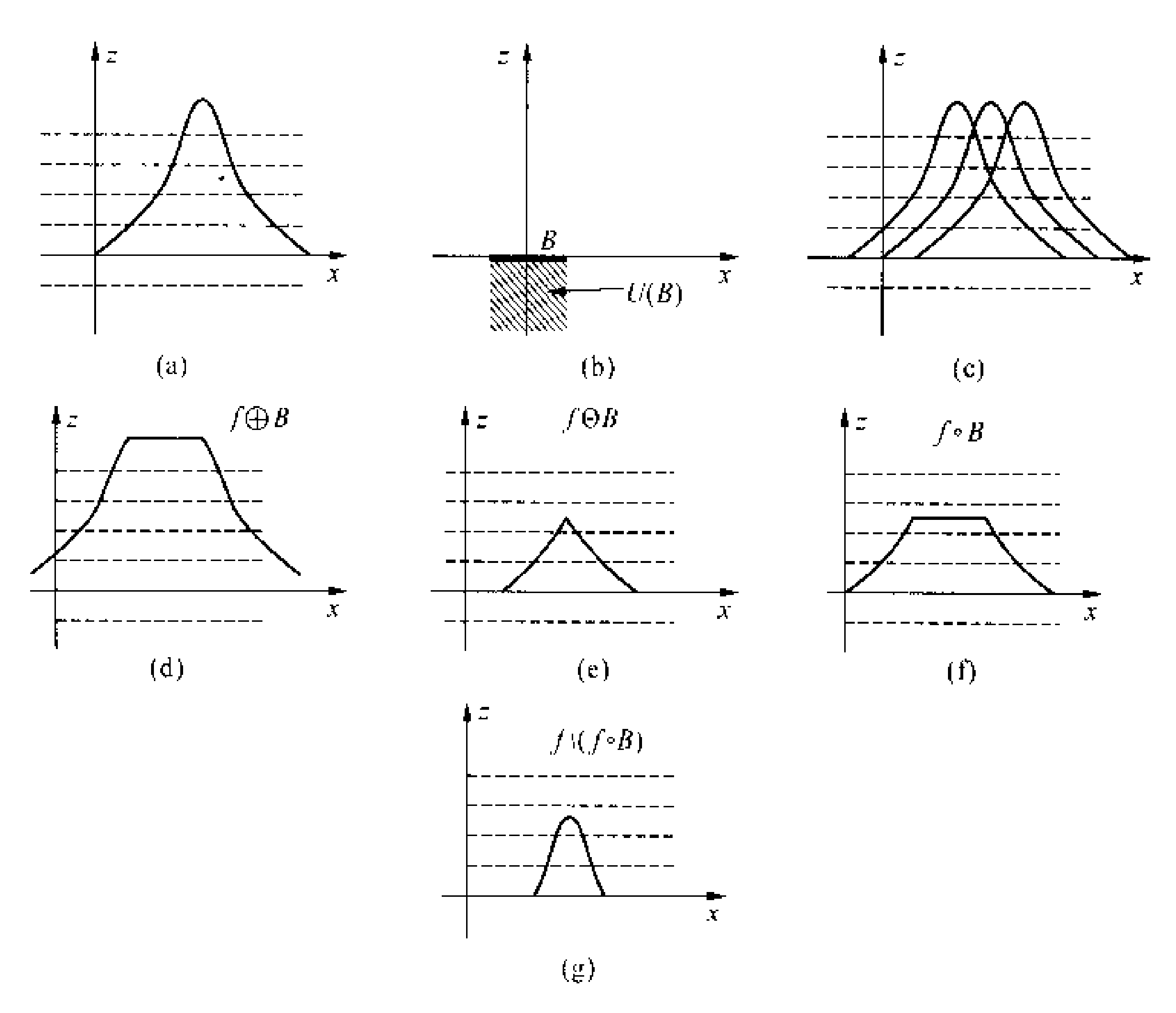
$$U(f \ominus g) = U(f) \ominus U(g) \qquad (Ⅱ-33)$$

并可知经过结构差运算后,(x, y) 点上的灰度为:

$$e(x, y) = \min_{(i,j)}(f(x+i, y+j) - g(i, j)) \qquad (Ⅱ-34)$$

在实际应用上常常遇到函数 f 与普通集合 B 之间的结构运算,如 $f \ominus B$ 和 $f \oplus$

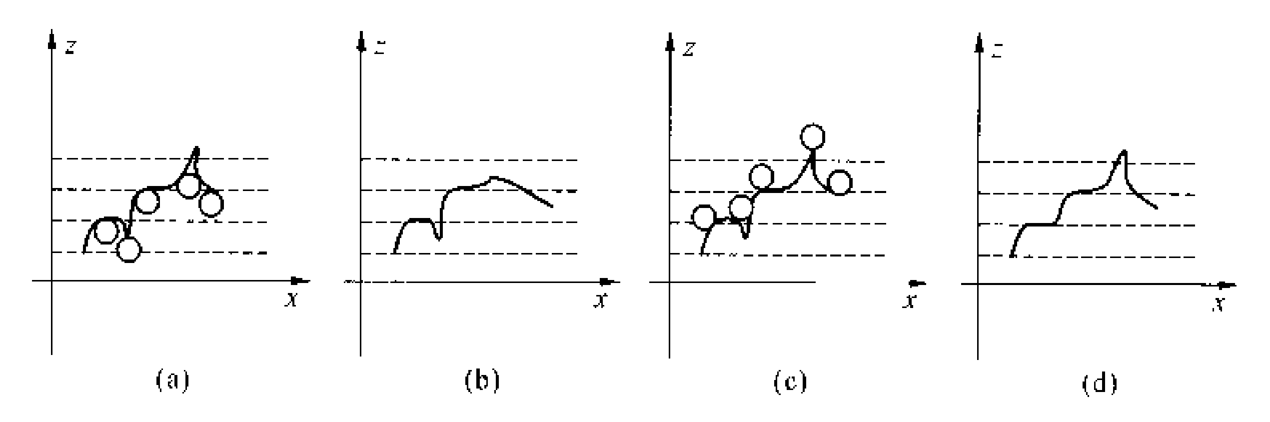
B 等，其中 f 为函数，B 为普通集合。由于普通集合 B 在增广了灰度维的空间仍可用 $U(B)$ 来表示，见附录图 12(b)，因此上面对 $f\Theta g$ 和 $f\oplus g$ 的定义及其计算公式可以完全适用于 $f\Theta B$ 和 $f\oplus B$。在附录图 12 的情况下，$f\oplus B$ 相当于将 f 平移以后取上包络，而 $f\Theta B$ 则相当于取下包络。在本例中，B 相对于原点对称，故函数 f 对普通集合 B 的结构开运算 $f\circ B=(f\Theta B)\oplus B$，这相当于先将 f 平移后取下包络，然后将下包络再平移，得到如附录图 12(f)的平顶图形，即磨去原图中的尖峰。而 $f\backslash(f\circ B)$ 则相应于取出山的顶峰，通常称 $f\backslash(f\circ B)$ 为高帽变换，见附录图12(g)。类似的有结构闭 $f\bullet B=(f\oplus B)\Theta B$，如果原来函数 f 有小凹缝，则 $f\bullet B$ 起填平凹缝的作用。



附录图 12　函数与普通集合的结构运算

参考文献[3]提出了函数 f 对球状结构元素 S 的结构开与结构闭的计算方法。这时结构开运算相当于用小球在函数曲面 f 的下面滚动。在 f 的小尖峰处，小球无法顶上去填满它，则经过这样的结构开运算后，这些尖峰就磨平了，见附录图 13(a)和(b)。而结构闭运算则相当于小球在函数曲面 f 的上面滚动，遇到小凹陷，小球落不进去，从而填平了凹陷，如附录图 13(c)和(d)所示。文献[3]还介绍了其

他形状结构元素的结构开和结构闭等内容。



附录图 13 函数 f 用球形结构元素 S 的结构开和结构闭运算

参考文献

[1] Serra J. Image Analysis and Mathematial Morphology[M]. Academic Press, 1982.
[2] Serra J. Introduction to mathematical morphology[J]. CVGIP, 35, 1986, 283~305.
[3] Sternberg S R. Grayscale morphology[J]. CVGIP, 35, No. 3, 1986, 333~355.